Modeling Responsibility Transfer In Man-Machine Systems

***S. Ahmad*[†], *M. A. Faruqi*[††] *and V. Saxena*[†††]**

† Azad Institute of Engineering & Technology, 226002 INDIA
††Azad Institute of Engineering & Technology, 226002 INDIA
†††Babasaheb Bhimrao Ambedkar Central University, 226002 INDIA

Abstract

While there is a continuing trend towards further automation and computerization of complex engineering systems to improve their performance and reliability, there is also an increasing realization that a visible over all human responsibility through monitoring and control of these systems particularly man-machine systems has to continue. This need for the intervention from human beings is acceptable largely due to their better over all situation perception and their crisis handling capabilities. The problems of automation and computerization in the man-machine domains have often been successfully addressed through modularization of the domain, as for example it can be seen in the case of Air Traffic Control (ATC) operations. This approach allows the designers to build an overall human control over each of the modules and it further facilitates in building human interaction and collaboration processes between the modules. Under these conditions the depiction and the visibility of responsibility as it is being shared and transferred between different human beings becomes an important modeling issue. It has been shown through the example of ATC that the above responsibility requirements can be modeled using the extensive massage handling and organizing features of the Unified Modeling Language (UML) Sequence Diagrams. This requires that the development of the diagrams should be carried in such a way that all responsibility sharing actors are included in it. Further, it is shown that through a proposed convention, both full and partial or shared responsibilities can be visually depicted on the diagrams. This approach also paves the way for modeling and depiction of human-computer or human-human interfaces that facilitate mediation between the computers and situation aware human beings.

Key words:
Air Traffic Control, Domain Driven Design, Human-Computer Interaction, Transfer of Responsibility

1. Introduction

Human beings have a unique sense of responsibility, commitment and overall situational awareness, which are difficult to build in the present day computers systems. Thus, they continue to play deciding roles in all safety critical systems specially those involving other human beings. The highly computerized and automated domain of Air Traffic Control (ATC) taken as an example here, is one such field where the overall decision making and control by human beings is an accepted practice. The Air Traffic Control services have evolved and matured over time and today their basic role is to give guidance to aircrafts to prevent collisions and to manage efficient traffic flow. However, to meet the ever increasing safety expectations the aviation systems have been going through a process of continuous improvement and change. These changes coupled with the increasing volume of air traffic, have lead to a significant increase in the work load of Air Traffic Controllers as well as that of the Pilots [1]. Normally, in an ATC system, various operational responsibilities pertaining to take-off, landing and flying of aircrafts are distributed between the pilots and controllers. Some of the responsibilities need to be routinely transferred between the pilots and the controllers while other may have to be done to re-distribute the workload of pilots and controllers as has been brought out in [2, [3], [4]. These new possibilities of responsibility transfers have emerged only recently with the increased capability and availability of relevant communication and control technologies.

Responsibilities [5] are essentially either causal which relates to make things happen or consequential which relates to answerability when things happen that should not. While causal responsibilities have same times been assigned to computers, the consequential responsibilities are almost always human and moral. While responsibility has a range of meanings in human-computer environment the moral aspect remains most relevant in the ATC context. Further, the view elaborated in European Commission's 'iFly' project [6] that '…a responsible person is defined as to who can take action if required without being required to request permission from another actor' has been adopted here for modeling.

In an effort to evolve the modeling of the responsibility in a man-machine system, roles of mediation and modularization have been brought out [7]. Under these conditions, the capabilities of UML (Unified Modeling Language) [8] Sequence Diagrams have been examined here for modeling some of these functional and non-functional requirements.

The presentation has been divided into five sections. Section 2 presents a brief literature review on responsibility modeling and the related issues in human-computer (man-machine) environment. Section 3 deals with the proposition of representing mediation,

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

virtualization and responsibility transfer modeling through extended UML Sequence Diagrams. Section 4 deals with the actual construction of the Sequence of Diagrams of the activities and responsibility transfers, for a portion of human-computer activities in ATC. The last section deals with the comments and the conclusion.

2. Literature Review

There has been a wide spread adaptation of 'Model Based' development technologies for large and complex IT based systems, due to their ability to capture relevant aspects of a system from any given perspective and also due to their ability to permit precise levels of abstraction. Further, these models allow automated development of the systems from their corresponding models along with various other model transformation possibilities and thus these features have led to a switch towards Model Driven Engineering (MDE) [9]. Work on MDE has largely been centered around the development and use of UML. UML has been particularly successful in presenting human understandable descriptions to both the domain experts as well as to software experts on various aspects of the system, which then could also be mechanically analyzed. Although, considerable portions of software development activities today tend to be domain specific, UML has not evolved to be able to provide [10] means for precisely defining semantics associated with these domains. However, as the UML based modeling efforts allow increased transparency and understanding between the different parties involved in the development and operation of the system, it helps in further improving the system design itself [11], [12].

With the increasing system complexity and safety requirements there has been a growing interest in the fields of human computer interaction like that of ATC, Unmanned Aerial Vehicles, surgical robotics etc. [13], [14], [26] towards improving their quality of performance. Presence of human beings in the control of these systems helps significantly towards providing reliability and a sense of human responsibility, which computers alone do not provide. It also helps in various other ways like that of providing a means of mediation between computer and its external environment [7], re-interpretation of data and in generation ideas for handling emergencies etc. Thus, the assignment of responsibility and its sharing is an important issue in human computer environment. It has been modeled in a variety of ways. An interesting approach towards responsibility transfer modeling between humans and computers as it shuttles from one entity to other in a chain like environment is shown by Huang and Katayama [15]. In their proposition, various aspects of the responsibility are modeled through an all-together different and parallel view of the system, away form the computation and control view.

The problem of responsibility transfer in the field of ATC has been studied by Friske et. al [2], [3] where in an effort to reduce the work-load on the ground controller, the responsibility of the maintenance of the separation between the aircrafts is proposed to be transferred to the air-craft pilots. A procedure involving all the concerned parties and an external agent has been evolved with the help of UML Sequence Diagrams. The general applicability of the proposition in human/human and human/machine environment has not been studied.

However, with the increasing need of further automation and implementation of new procedures towards increasing the safety, reducing travel time and fuel consumption of aircrafts etc. more studies are needed to evolve the required responsibility transfer ideas and procedures in the mixed human computer environment. Zemrowski [16] has even observed that many of the above mentioned objectives may be implemented through technologies like 4D path specification, enroute trajectory negotiation and enroute free maneuvering etc., which will require increased degree of human computer interaction.

Further, with the increasing reliance on computers (without transferring the over-all responsibility to computers) in the fields like that ATC field, the Human Computer Interaction can now be said to be more of Human Computer Collaboration [17], [18]. It has also been suggested that such collaborations should possibly have an agreement between humans and computers on the shared goals, the allocation of responsibility, the ability to track progress, the capability to negotiate and the adaptation to situation etc. As these capabilities are difficult to invoke on the computer side, the locus of responsibility remains with the human beings.

The above literature survey points to the fact that while there has been considerable research activity towards understanding the nature of human responsibility and its perception, there is only a limited information available about its modeling as a functional or non-functional requirement of activities. A UML based modeling approach, started by Friske et. al. which is suited to virtualized (networked) environments, has been adopted here. Efforts have been made to show that this approach can be generalized and used for modeling the continuity of human responsibility in real-world situations. Efforts are further made to show that these UML models can also be used for the abstraction and representation of responsibility at different levels of details.

3. Modeling Responsibility in Human-Computer Environment

Most Human-Computer interaction environments have imperatives imposed by the particular domains. As pointed out earlier, for example, the safe and re-assuring operational capability of ATC domain operations has largely been achieved through the modularization of the domain and the complexity and through making humans as the responsible and visible end of the sub-domains (or modules) [19]. Such a layout of the pre-take-off (ground) control of aircrafts through a series of controllers is shown in Fig.1 from reference [20].

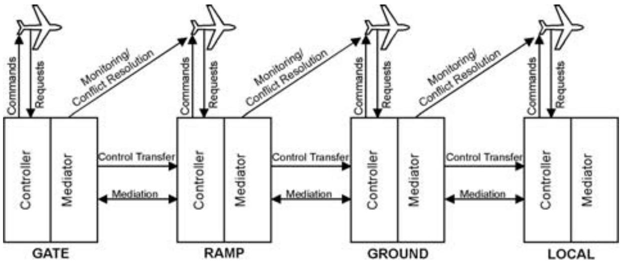

Fig. 1: A layout of the interconnected pre take-off controllers at a typical airport, showing the involvement of human being at each step

Based on this approach, requiring human visibility for responsibility, it is proposed here that this can be generalized and activities around each human being may be abstracted and modeled as a unit as shown in Fig. 2. It may comprise of two parts, one consisting of networked computers with a front end computer, and the other consisting of the human being, with a human computer interface between the two. In this way the available information may be mediated by human beings before the action is initiated. Further, mediation between any two modules may also be done by human beings as shown in Fig. 3.

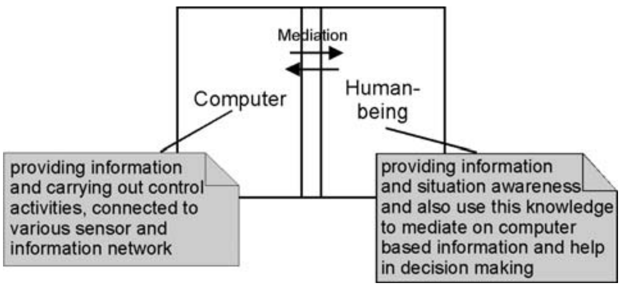

Fig. 2: A module creating a sense of responsibility through mediation

It can also be seen from Fig. 3 that in this model, as is required, the overall control and thus the responsibility remains in the hands of human beings and wherever the automation fails or the information needs interpretation, human beings are available, in spite of very high degree of automation. Human beings in this interpretation are thus cast into two different roles. One is that of bringing in the overall awareness information and the other is that of mediation between this information and the computerized information. Such a dual role is possible in human computer interaction situations, as shown by Bodker and Andersen [7].

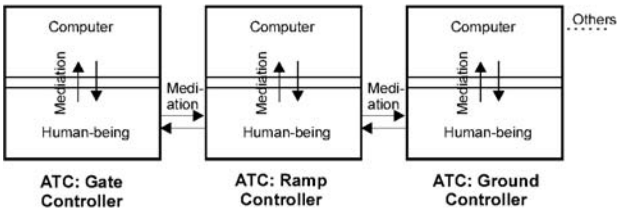

Fig. 3: Human-computer and human-human mediation between controllers in a modular system

Using these modules as building blocks, it can be seen that an overall pattern of responsibility and its transfer may be modeled in a human-computer system.

As pointed out earlier, with the continuously increasing air traffic, there is a growing need for increased levels of automaton and reliability of the ATC operations at each of the stages. These requirements are further coupled with the new requirements of implementing fuel saving procedures [16], and the need for pilots involvement in the aircrafts separation maintenance etc. Some new directions of solutions have also merged, particularly through the virtualization that is making data and model available anywhere on the computer network. For example in term of the modules proposed here (Fig .3) any of the modules along with its data can be developed in such a way that it can be controlled from anywhere on the network or can be collapsed (or merged) with any of the other modules. While there may be no problems in adding the extra load on the computer systems, the practicability of the move will be determined by the human being's capability of handling the joint modules and also having the necessary awareness to carryout both the roles. Multiple roles for human actors are common in the real-world.

It is thus proposed here (detailed in next section) that by separating the human and the computer roles in each of the modules and by separately providing communication between the human parts of the modules, the modeling of human responsibility can be carried out on a continuous basis. This type of modeling can be accomplished through UML Sequence Diagrams, if each of human parts of the modules is represented as an actor in the diagram.

Further as, the UML Sequence Diagrams allow the depiction of real-time operations and the identification of control actions, it may help in modeling of the responsibility.

4. Modeling the Responsibility Transfer through UML Sequence Diagrams

In this section, the transfer of responsibility in a human-computer system has been studied through first evolving the message and the responsibility transfer protocols and then by using these for constructing UML Sequence Diagram involving all the human actors. Next a procedure for depicting the responsibility and its transfer on the diagram is evolved.

4.1 Transfer of Responsibility Protocol

In general in a man-machine system, various responsibilities may be distributed between the actors (men and machines) participating in the particular system. But as outlined earlier the consequential responsibility is important and it has to be carried essentially by human beings only, thus its transfer between human beings (actors on the diagram) needs to be modeled. It may be assumed that system has been designed in such a way that these human actors have a degree of control over all the operations, including the automated portions at all times.
Depending upon the need [15] the transfer of the responsibility between the actors has to precede with negotiations, using a transfer of responsibility protocol between the actors which can be depicted on a UML Sequence Diagram like the one shown in Fig. 4.

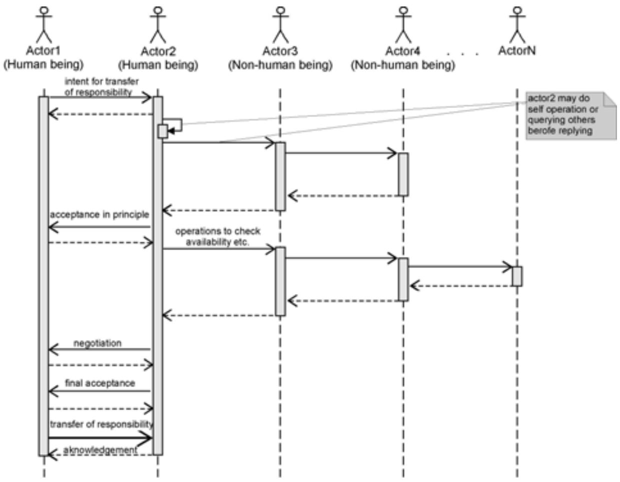

Fig. 4: Sequence Diagram showing negotiations before the transfer of the responsibility between the human actors of a system

The figure shows negotiations for the transfer of responsibility between two actors and allows a possible involvement of others in this process. If in the beginning the responsibility is with the Actor1 and after performing its task when it wants to transfer the responsibility to the Actor2, then it conveys its intent to the Actor2 about the transfer. The Actor2 then sends a response message to the Actor1 indicating whether it is interested in accepting the transferred responsibility or not. Actor2 then may do self-operation or query others before proceeding further. Based on the Actor2's response whether it is willing or not to accept the transferred responsibility, various other negotiations may be done between the Actor1 and the Actor2. The Actor1 finally transfers the responsibility only when the Actor2 is ready to accept the responsibility and further when the transfer is complete, Actor2 sends an acknowledgement to Actor1 that it has successfully accepted the responsibility. The responsibility transfer can thus be modeled in a similar way between any two of the participating actors of the system.

4.2 Modeling the Transfer of Responsibility

As discussed earlier, since the responsibility has to be carried only by human beings the UML diagram has to be constructed with all the actors who carry full responsibility at one time or another. Further, the instances of shared responsibility between the main actor and the contributing actors have also to be identified.

Continuing with the example of the responsibility transfer modeling for ATC (moments of the aircraft before take-off), it can be seen [20], [21] that there are five actors (humans) who exercise full responsibility at one time or another. They are the Gate Controller, the Ramp Controller, the Pilot, the Ground Controller and the Local Controller and thus they have been be used as actors for developing the UML Sequence Diagram.

The consequential human responsibility [22] being modeled here in fact becomes more like the operational responsibility [23] between the pilot and the controllers for piloting the aircraft so that a safe and efficient operation of the flight takes place. The responsibility can further be either full or can be shared with any of the other actors.

The negotiations, as mentioned earlier, have to precede before the transfer of the responsibility takes place. The transfer of the full responsibility of an actor on the Sequence Diagram is proposed to be indicated by a thick solid arrow with a dark rectangular (■) mark at the end. The shared responsibility is proposed to be indicated by a hollow rectangular mark (□). The controller who has the full operational responsibility remains the main controller and the controllers who have the shared operational responsibilities just keep monitoring and giving instructions, if they feel that they are required. The pilot and controllers communicate through the existing and established communication networks and protocols [24].

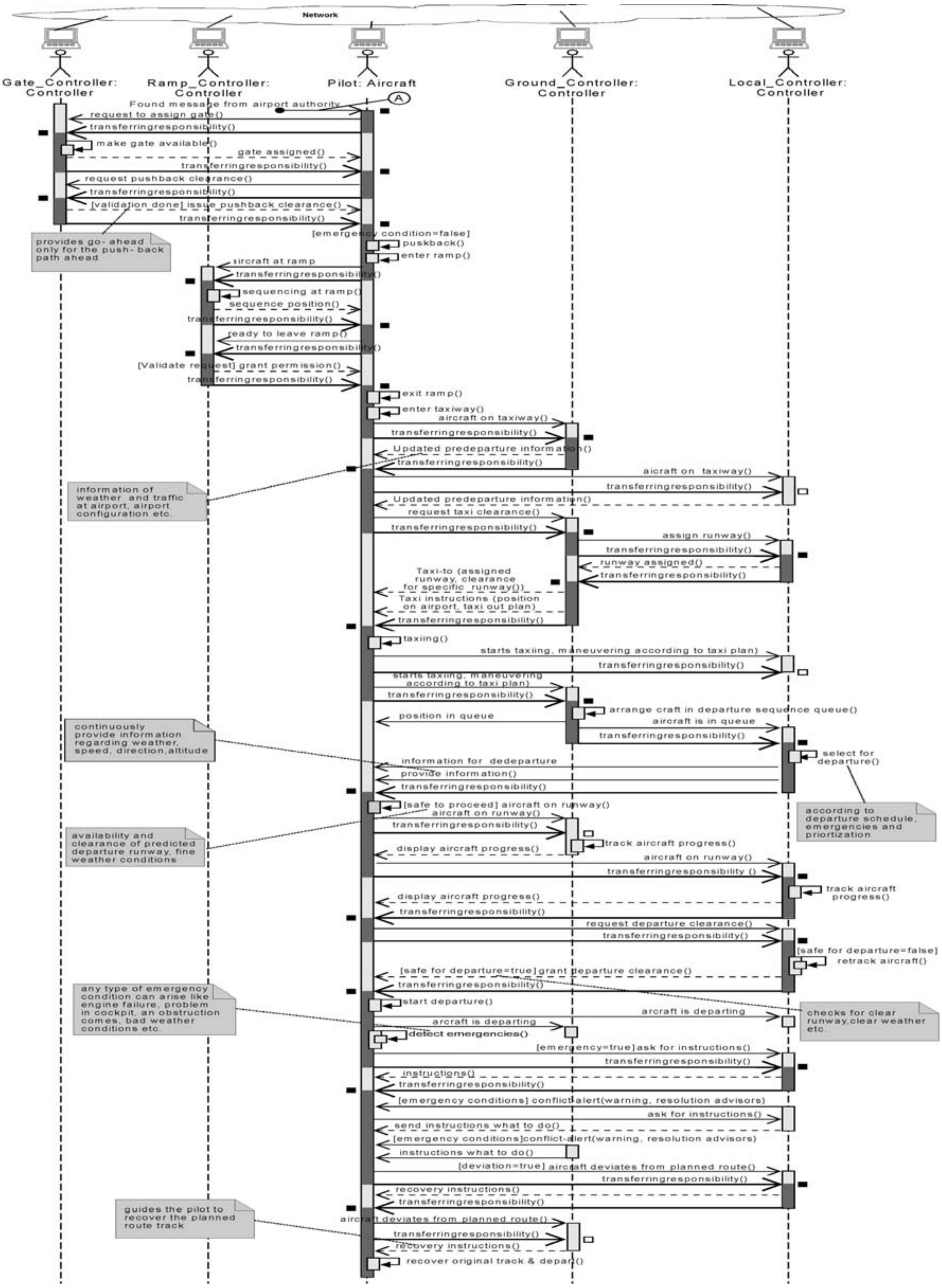

Fig. 5: The UML Sequence Diagram with imposed the Responsibility Transfer Diagram for the departure of an Aircraft

The proposed method of depicting the responsibility and its transfer is carried out in two steps. First, a sufficiently detailed (at a level at which individuals hold and exercise responsibility) UML Sequence Diagram is prepared keeping the requirements mentioned above in focus. Then on this diagram, the transfer of responsibility is depicted using the conventions mentioned above.

4.2.1 Constructing a Sequence Diagram for Depicting the Responsibility Transfers during the Departure of an Aircraft

The Sequence Diagram (Fig. 5) for the departure of an aircraft in the ATC environment is constructed between the five human actors as identified in the previous section. One of the actors is always in command of the operations and also serves as mediator between the computerized information and the real-world awareness. The conventions used for the construction of the diagram are as described in the References [13] and [25]. The solid lines represent the call messages and the dotted lines represent the message returned between the actors. Tags have been used to improve the readability of the diagram and guards have been used at the decision points.

The process starts from a point marked as 'A' in the diagram, when the pilot receives a message from Airport Authority clearing its departure request. This clearance information is made visible to all other actors as well.

The Sequence Diagram constructed is largely self explanatory and is developed from the sequence described in [20] for the departure of an aircraft. The pilot (after receiving the departure message) requests the Gate Controller to assign a gate. The Gate Controller then performs a self operation to make the gate available and sends a message to the pilot informing about the assigned gate. The pilot then brings the aircraft up to the gate and requests the Gate Controller to grant clearance for pushback from the gate. The Gate Controller then issues a pushback clearance, after performing the necessary validations. If there are any emergency conditions then a check-emergency operation is called recursively, else the pilot pushes back the aircraft from the gate and enters the ramp area. Now the pilot informs the Ramp Controller that the aircraft is at the ramp. The Ramp Controller does the necessary sequencing of the aircrafts at the ramp and sends a message to the pilot about the position of the aircraft in the queue. At his tern the pilot sends a message to the Ramp Controller, asking it to grant the permission to leave the ramp. The Ramp Controller then does necessary validation of pilot's request, and grants the permission to the pilot to leave the ramp. The process continues in this way till the aircraft takes-off.

4.2.2 Constructing the Responsibility Transfer Diagram over the Sequence Diagram for the Departure of an Aircraft

The Sequence Diagram evolved in Fig.5 is used further to develop and depict the conditions and states under which the transfer of responsibility takes place between the pilot and the various controllers during the departure of the aircraft. This may be accomplished through messaging and the use of the responsibility transfer protocol. As can be seen, the transfer of responsibility takes place at several instances during the whole departure activity. Initially, the full responsibility is with the pilot (marked with a filled square (■)), who on receiving a message from the Airport Authority regarding the permission for departure, transfers the responsibility to the Gate Controller for further actions. This is shown by a thick line in the diagram with a filled square (■) at the end. After assigning the gate, the Gate Controller transfers the responsibility back to the pilot. The pilot now acts and brings the aircraft to the gate and transfers the responsibility back to the Gate Controller by requesting for the granting of the clearance for pushback from the gate. The Gate Controller then issues the pushback clearance and transfers the responsibility back to the pilot. Now the aircraft moves again and enters the ramp area and the pilot transfers the responsibility to the Ramp Controller. Now when the Ramp Controller grants the permission to the pilot to leave the ramp, it transfers the responsibility back to the pilot.

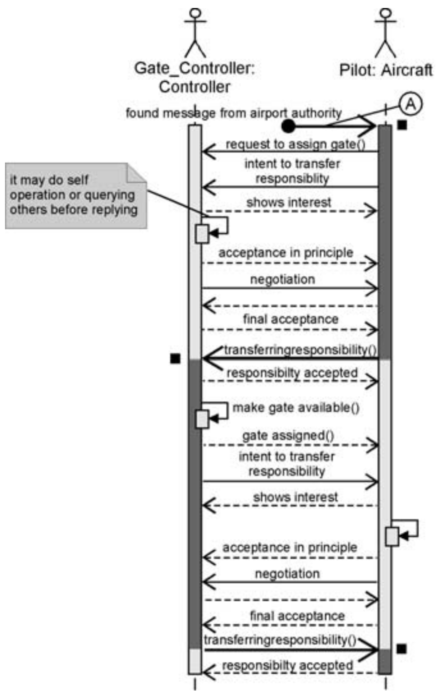

Fig. 6: Sequence Diagram showing the transfer of responsibility during bringing of an aircraft to the gate for departure

A situation of shared responsibility occurs when the aircraft is on the taxiway and the pilot transfers the full responsibility to the Ground Controller and a shared responsibility to the Local Controller. The shared responsibility in this proposition is said to occur, when more than one human actors (in terms of UML Sequence Diagram) are involved at the same time, in a cooperative way, for carrying out an activity.

The further depiction of the responsibilities on the diagram can be continued as shown in Fig.5.

The departure activities of an aircraft as modeled in the Sections 4.2.1 and 4.2.2 have been abstracted at a level to be able to mark the transfer of responsibility points in the Sequence Diagram (Fig.5). Further details in the model can be provided by going to the negotiations level or further lower to the message transfer levels as has been shown in Sequence Diagram in Fig.6 for the initial portions of the operations.

5. Conclusion

With the increasing complexities in the man-machine systems, ethical and moral standards today demand that human beings should be in overall control, to provide a human sense of responsibility to the system and to help in increasing the dependability through their situation aware inputs. This requires that system design of the man-machine systems should be modularized in such a way that the involved human beings are able to comprehend the area of their control.

Using ATC operations as an example, where extensive modularization exists and responsibilities are defined through the human beings in the system, it has been shown in the studies here that:

1. The UML Sequence Diagrams can be adapted for the modeling and the displaying of human responsibility in man-machine systems and for its transfer from one human actor to another in the form of a continuous chain, as the system operations are carried out.
2. The proposed responsibility modeling UML Sequence Diagrams may be drawn at different levels of abstractions. At the lower levels it may show the other parties involved and the messaging necessary to accomplish the transfer.
3. The graphical representation of the actors holding the responsibility at any time and its transfer as the process continues may help the stake holders in having a better appreciation of their roles and may further help in producing more robust computer codes.

References

[1]. Gerald L. Dillingham, “AIR TRAFFIC CONTROL: FAA Needs to Better Prepare for Impending Wave of Controller Attrition,” a Report to the Chairman and Ranking Democratic Member of the Subcommittee on Aviation, House Committee on Transportation and Infrastructure, United States General Accounting Office, Jun 2002.

[2]. M. Friske, R. Ehrmanntraut, "Modeling a Responsibility-Transfer Service for The Delegation of Tasks in Airborne Separation Procedures and Ground Automation,” Aerospace Science and Technology 9, pp.533–542, 2005.

[3]. M. Friske, “Modeling a Responsibility-Transfer Service,” DGLR-Jahrbuch, Deutsche Gesellschaft fuer Luft-und Raumfahrt - Lilienthal-Oberth eV, Bonn, 2002.

[4]. L. Brim, “Fundamentals of Air Traffic Control,” ParaDise Seminar, Feb 2008, Available: http://www.fi.muni.cz/paradise/Seminar/2008-spring/ 0225-prezentace.pdf

[5]. J. Dobson, “Responsibility Modelling: Basic Concepts,” Responsibility and Dependable Systems, Springer London, pp. 91-114, 2007.

[6]. A. Luuk, J. A. Wise, F. Pouw and V. Gauthereau, “iFly Deliverable D2.1: Description of airborne human responsibilities in autonomous aircraft operations,” Dec. 2007. Available: http://ifly.nlr.nl/ documents/D2.1_iFly_Final_1.2_Dated%2028_Dec _2007.pdf

[7]. S. Bodker, P. B. Andersen, "Complex Mediation", Human–Computer Interaction, Lawrence Erlbaum Associates, Inc., Volume 20, pp.353–402, 2005.

[8]. G. Booch, J. Rumbaugh, I. Jacobson, “The Unified Modeling Language User Guide,” Addison-Wesley, Reading, MA, 1999.

[9]. R. France, B. Rumpe, “Model-driven Development of Complex Software: A Research Roadmap,” Future of Software Engineering, IEEE Computer Society, 2007.

[10]. R. B. France, S. Ghosh, T. Trong, A. Solberg, “Model Driven Development Using UML 2.0: Promises and Pitfalls,” IEEE Computer Society, pp.59-66, Feb. 2006.

[11]. S. Sendall, “Domain-Driven Software Development—A World of Transformations,” Proceedings of the 5th IEEE International Workshop on Rapid System Prototyping, IEEE Computer Society, 2004.

[12]. E. Evans, “Domain-Driven Design: Tackling Complexity in the Heart of Software,” Addison-Wesley, 2003.

[13]. C.D. Wickens, A.S. Mavor, R. Parasuraman, J. P. McGee, “The Future of Air Traffic Control: Human Operators and Automation,” National Academies Press, Jan 1998.

[14]. R. Prasuraman et. al., "Adaptive Automation for Human-Robot Teaming in Future Command and Control Systems," the International C2 Journal, vol. 1, no. 2, 2007.

[15]. M. Huang, T. Katayama, "Steering Model-Driven Evolution by Responsibilities," Proceedings of the 8th International Workshop on Principles of Software Evolution, IEEE Computer Society, 2005.

[16]. K.M. Zemrowski, "Impact of Increasing Reliance on Automation in Air Traffic Control Systems," IEEE International Systems Conference, SysCon, 2008.

[17]. P.U. Lee, "Understanding Human-Human Collaboration to Guide Human-Computer Interaction Design in Air Traffic Control," IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 1598-1603, 10-12 Oct. 2005.

[18]. G. Cox, S. Sharples, A. Stedmon and J. Wilson, "An observation tool to study air traffic control and flightdeck collaboration," Applied Ergonomics, Vol. 38, no. 4, pp.425-435, July 2007.

[19]. B. Shneiderman, "Human Responsibility for Autonomous Agents," IEEE Intelligent Systems, IEEE Computer Society, 2007.

[20]. R.J. Hansman, "Air Traffic Control Overview," MIT Department of Aeronautics and Astronautics, MIT ICAT, 2006. Available: http://ocw.mit.edu/NR/rdonlyres/Aeronautics-and-Astronautics/16-72Fall-2006/11CD67DE-29C1-4AB9-A7DD-004BB1897CB9/0/lec1.pdf

[21]. I. Anagnostakis, H. R. Idris, J. P. Clarke, E. Feron, R. J. Hansman, A. R. Odoni, W. D. Hall, "A Conceptual Design of A Departure Planner Decision Aid," 3rd USA/Europe Air Traffic Management R&D Seminar, June 13-16, 2000. Available: http://dspace.mit.edu/bitstream/1721.1/37321/1/paper68.pdf

[22]. "Computing and Moral Responsibility," Standard Encyclopedia of Philosophy, Aug. 2004.

[23]. A. Krauter, "Roles and Responsibilities in Weather Decision Making: The Dispatcher's Perspective," Available: http://onlinepubs.trb.org/ Onlinepubs/ circulars/ec034/krauter.pdf

[24]. "Air Traffic Control Complete and Enforced Architecture Needed For FAA Systems Modernization," United States General Accounting Office - Report to the Secretary of Transportation, GAO/AIMD-97, Feb. 1997, Available: http://www.gao.gov/archive/1997/ai97030.pdf

[25]. D. Bell, "UML's Sequence Diagram," IBM, May 2008,Available:http://www.ibm.com/ developerworks/rational/ library/3101.html

[26]. R. H. Taylor, "Medical Robotics and Computer-Integrated Therapy Delivery: Coupling Information to Action in 21'st Century Therapeutic Interventions Century Therapeutic Interventions," CISST ERC, 2000. Available: http://www.cs.jhu.edu/~rht/Talks/CIS-NSF-ASME-Shortened.pdf.

Shafeeq Ahmad received his Masters degree in Computer Applications from Madurai Kamaraj University, India in 2002. He is working as an assistant Professor in the department of Computer Applications at Azad Institute of Engineering & Technology, Lucknow, India. His research interest includes Software Engineering and Human-computer Interactions.

M. A. faruqi received his M.E. and Ph.D. degrees from Indian Institute of Technology, Kharagpur, India in the years 1965 and 1974 respectively. He has been a Professor at IIT, Kharagpur and has worked in the areas of CAD/CAM, Robotics and Neural Networks. Currently he is at Azad Institute of Engineering & Technology, Lucknow, INDIA.

Vipin Saxena is a Reader and Head, Dept. of Computer Science, Babasaheb Bhimrao Ambedkar University, Lucknow, India. He got his M.Phil. Degree in Computer Application in 1991 & Ph.D. Degree work on Scientific Computing from University of Roorkee (renamed as Indian Institute of Technology, India) in 1997. He has more than 12 years teaching experience and 16 years research experience in the field of Scientific Computing & Software Engineering. Currently he is proposing software designs by the use of Unified Modeling Language for the various research problems related to the Software Domains & Advanced Computer Architecture. He has published more than 55 International and National publications.