

A Modeling and Detection of Deadlock in Early Stage of System using UML

Gufran Ahmad Ansari
College of Computer,
Department of Information Technology
Qassim University, Saudi Arabia.

ABSTRACT

As we know there is an increasing demand from the software industry to develop software model which can communicate and exchange information concurrently in multiprogramming environment. This is very difficult to find and eliminate concurrency problem like deadlock in a large and complex system. There is a need of model to identify and recognize concurrency problem such as deadlocks in the early stage in system and design model to get rid of these kinds of problems. In this paper author proposed a UML model for detecting the concurrency in early stage of design system. The author has also proposed a sequence diagram, activity diagram and use case diagram for the above model.

Keywords

UML class model, Deadlock detection, Software, Concurrency

1. INTRODUCTION

Unified Modeling Language (UML) is a well-known software modeling language and it is accepted by the academia as well as by the software industry people. It has also become a standard language for software analysis and design [1]. The modeling is an advance way of thinking about the scientific research problems in a well organized manner in real-world ideas. Some important model systems are designed and explained in [2, 3, 4,]. UML is a combination of good engineering practices that has proved to be successful in modeling a software design of difficult or complex systems. Modeling is very helpful to understand the process of input and output of system [5]. Recently, Saxena & Ansari proposed a different UML models which are based on specific domain [6, 7, 8]. These models are very helpful to understand the “UML modeling and design concepts”. A lot of literature is available on modeling problems by the use of UML but limited research papers are available in literature to finding the deadlock in multiprogramming environment through UML applications. The deadlock is situation where a process or set

of processes is blocked, waiting for some resources that is held by some other process [9].

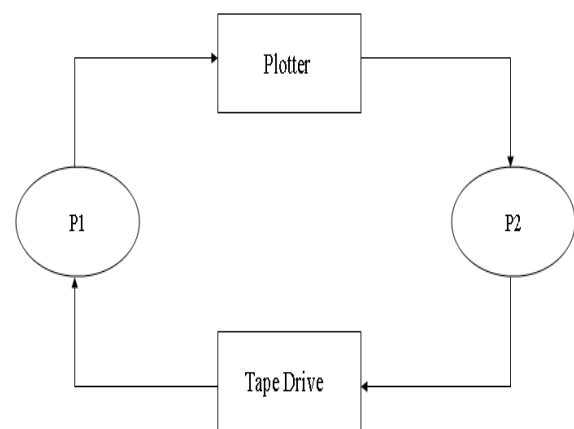


Fig. 1 Deadlock Situation

The Fig. 1 shows the situation of deadlock. In this figure system takes with a tape drive and a plotter. Let process P1 request the tape drive and process P2 request the plotter, both the processes receive these resources, as both the resources were initially free. Hence leaving these resources occupied process p1 request to the plotter and process p2 requests the tape drive. In this situation both the processes were not granted with the resource and both the processes goes in an indefinite wait and deadlock. The deadlock happen when a thread is not capable to continue its execution as it is blocked waiting for a lock that is held indefinitely by another thread [10, 11]. A thread is a flow of execution through the process code, with its own program counter, system register and stack. Threads are a popular way to improve an application performance through parallelism.

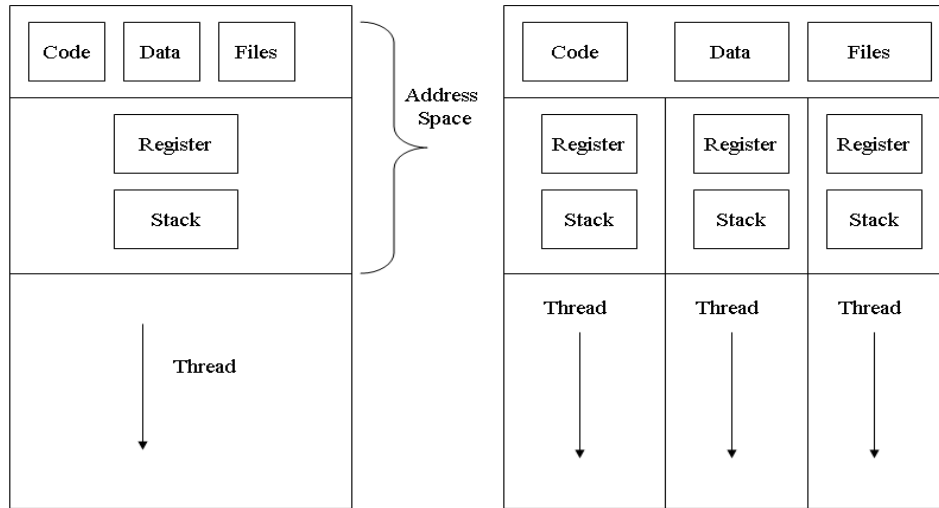


Fig. 2 Single and multi-thread processes

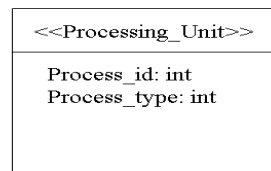
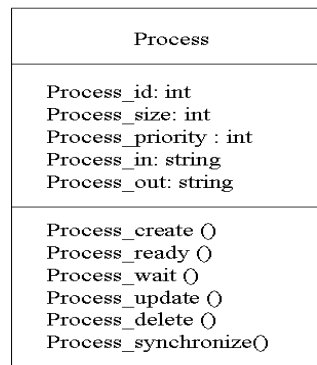
A thread is sometimes called as light weight process. Process is a dynamic entity that is a program in execution. A process is a sequence of instruction execution. Concurrency introduces the need for communication between executing threads, which is need to synchronize their operations and shared memory communication, which ensures that the shared resources are accessed individually and appropriately [12]. Monitoring and authentication usually consists of examination for communication faults such as deadlocks, infinite loops,

lovelocks and other communication pitfalls. As model-based software development practices are getting more popularity [13, 14], resource and communication deadlock models are considered in message communication systems. Most deadlock models in distributed systems are resource models [15, 16].The author proposed a model for the detection of concurrency problems that is based on design models expressed in UML [17].

1.1 UML Process

Process can be explained in the form of collection or block of instructions of program, macro subprogram or subroutine. To explain this process there is a need of processing elements.

The processing element are defined as stereotype and generally used to handle the concurrent processes. The given Fig. 3 shows the UML process and the processing unit.



(a) UML Process

(b) Processing Unit

Fig. 3 UML Process & Processing Unit

2. UML CLASS MODEL FOR DETECTION OF DEADLOCK

The class model for detection of deadlock in a system is designed with the help of UML concepts and is shown below in Fig. 4. The complete system for detection of deadlock is

shown with designed attributes and functions. The UML model contains the six major classes which are Process, OS Manager, Pre_Scheduler, Process_Exe, Check_Deadlock and Remove_Deadlock.

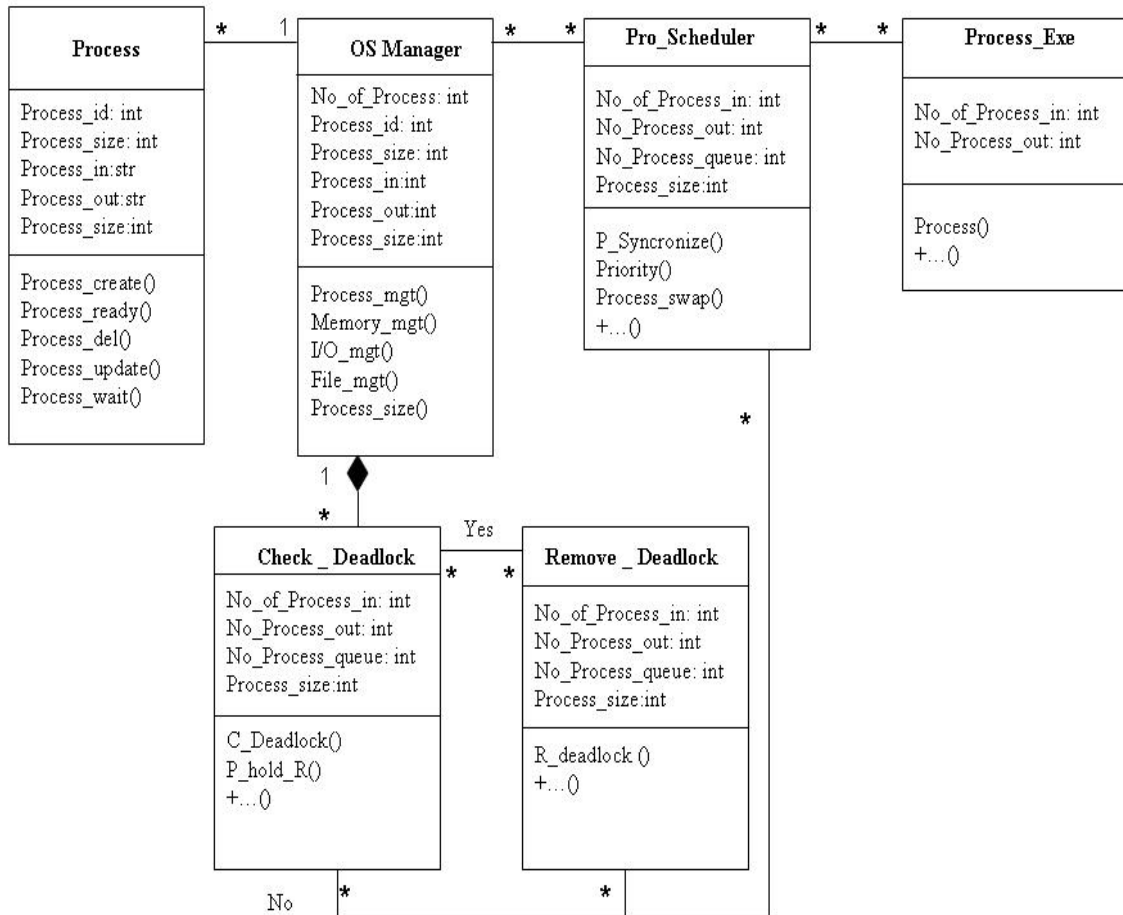


Fig. 4 UML Class Model for Deadlock Detection

The Process class is associated with OS Manager. Process class has multiple associations with OS Manager and OS Manager has a single association with the Process. Similarly OS Manager Class has a multiple association with Pre_Scheduler class and pre_Scheduler also have a multiple association with OS Manager. Check_Deadlock and Remove_Deadlock classes inherit from OS Manager. OS Manager Class has a single association with Check_Deadlock and Remove_Deadlock classes while Check_Deadlock and Remove_Deadlock classes have multiple associations with OS Manager Class. Check_Deadlock and Remove_Deadlock classes also have multiple associations. The Check_Deadlock and Remove_Deadlock classes also have multiple associations with Pre_Scheduler class and Pre_Scheduler classes also have multiple associations with Check_Deadlock and

Remove_Deadlock class. Process_Exe class has a multiple association with Pre_Scheduler class and Pre_Scheduler class also has multiple associations with Process_Exe class.

3. UML ACTIVITY DIAGRAM FOR DETECTION OF DEADLOCK

An activity diagram is a self-motivated diagram that shows an activity and an event. The activity diagram shows the variety of activities step by step with the movement of both conditional and parallel behaviors. Basically the activity diagram is the modification of state diagram in which most of the states are activities. The activity diagram of the above UML class model for detection of deadlock in a system is given in Fig 4.

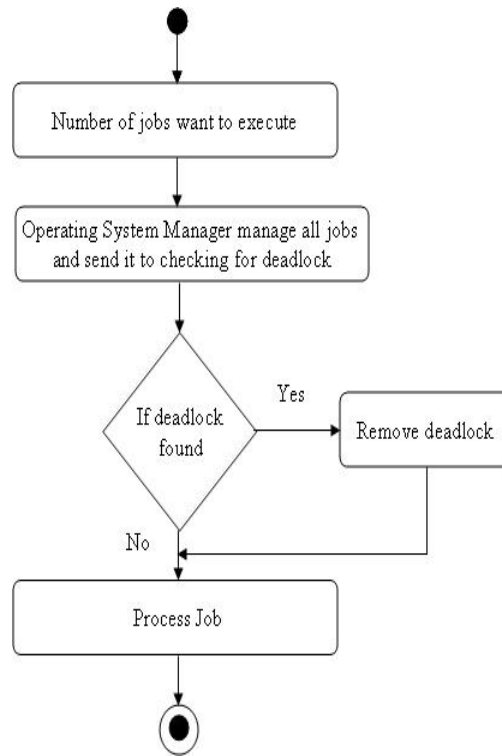


Fig. 4 UML Activity Diagram for Deadlock Detection

According to the activity diagram a number of jobs want to run in a CPU. The operating systems manage all jobs which are willing to run. The operating system manager first check for the deadlock, if deadlock is found then first it removes deadlock and wait till the processes are not removed from resources which are holded by the other process. If the process is not holding any resource of the other process, then the process processed jobs and wait for the next job.

5. SEQUENCE DIAGRAM FOR DETECTION OF DEADLOCK

A sequence diagram is made up of objects and messages. The given sequence diagram below show the detection of deadlock in a system. It has five major objects which are shown on the top of the diagram through the rectangular box with the underlined class name. The five objects are jobs, OS Manager, Deadlock, Remove deadlock and CPU Process Jobs.

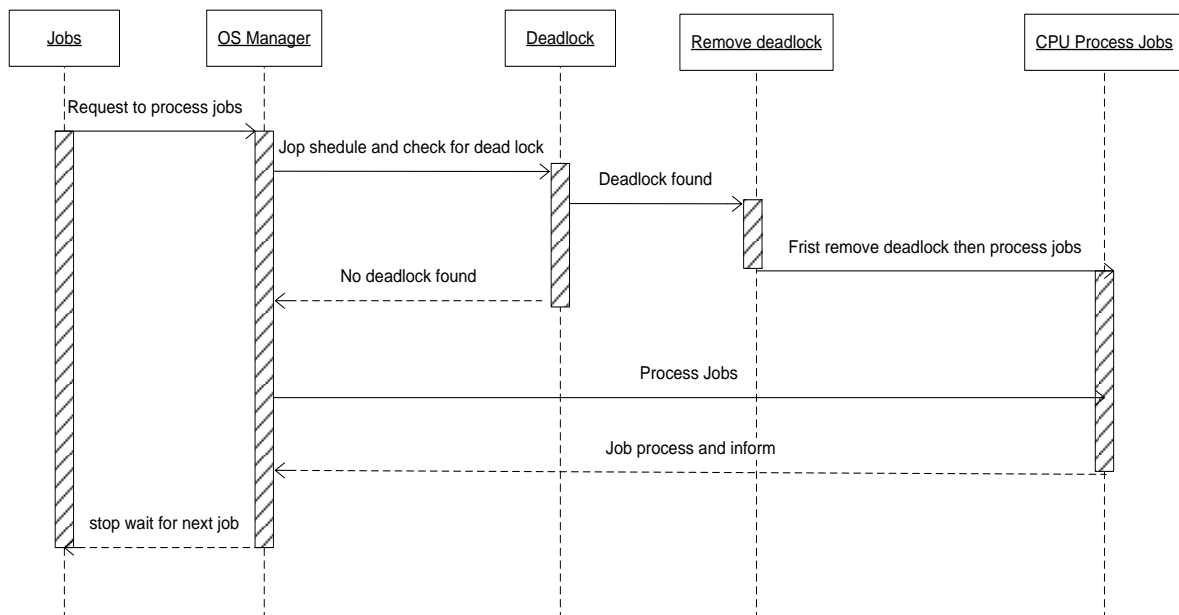


Fig. 5 UML Sequence Diagram for Deadlock Detection

The communication of information between two objects is represented by an arrow and the messages on that arrow. In a sequence diagram an object is shown at the top of the diagram with underline and vertical lines shows the life of the object. According to the sequence diagram a number of jobs requests OS Manager to execute? OS Manager schedules all jobs and checks for the deadlock. If any process is holding the resources of the other process, then first free the resources of other processes and then jobs can be executed other wise it can not be executed, it goes in deadlock situation. If process is not holding the resources of other jobs, then OS Manger sends to execute a job and waits for the next jobs.

6. CONCLUSION

From the above work it is concluded that UML is a powerful modeling language which is used to represent the scientific research problem. In this paper author has done a complete modeling for finding out the deadlock situation in a system. The UML model is very efficient and useful for the software developers who develop high quality software's and avoids the deadlock situation in a system. This UML model is flexible and easily extendible.

7. ACKNOWLEDGEMENTS

Author is very grateful to Prof. Dr. Khaled Bin Abdul Rahman Al Hamoudi, Rector of the Univesity and Dean College of Computer Qassim University Saudi Arabia for providing the excellent and tremendous research facility in the College of Computer.

8. REFERENCES

- [1] G., Booch, J.Rumbaugh, and I Jacobson. 1999. "The Unified Modelling Language User Guide", Addison-Wesley, Reading, MA.
- [2] Grady Booch. 1994. "Object-Oriented Analysis and Design with Applications", second Edition, Addison-Wesley.
- [3] Graig Larman 1998. "Applying UML and Pattern" Prentice Hall.
- [4] Ansari, G.A., "A Domain Oriented Modeling of Indian Education System through UML" the Icfai Journal of Systems Management (IJSJM) (ICFAI Press India), Vol. VIII; No.3; August, 2010.
- [5] Lieberherr, K., Holland, I., and. Rie, Al. "Object-oriented programming: An objective sense of style", Proc. OOPSLA'88 as ACMSIGPLAN 23, 11, pp.323–33.1198.
- [6] V. Saxena and G. A. Ansari, "UML Modeling & Protection of Domain Based System", International Journal of Computer Science and Network Security (IJCSNS), Vol. 8, No. 7, pp.338-344.,2008.
- [7] V. Saxena and Ansari, G.A. "UML Modeling for Patient Registration System" International Conference on Quantitative Methods, Operations and Information Technology for Managerial Decision Making ICQMOIT-2008, 24-25 Oct. at IBS, Hyderabad, India,2008.

- [8] V. Saxena and Ansari, G.A., Ajay Pratap. "Enhancing Security through UML", International Journal of Computer Sciences, Software Engineering and Electrical Communication Engineering". Vol.2,(1), pp 31-36, 2011.
- [9] A. Silberschatz, et. all 2008. "Operating System Concepts". John Wiley India Pvt., Ltd. 6th edition.
- [10] Downey, A. B.2008. The Little Book of Semaphores., Green Tea Press, 2nd Edition.
- [11] Crowley. 2006 "Operating Systems: A Design-Oriented Approach" Tata McGraw – Hill.
- [12] Pender, T.2003 UML Bible, Wiley.
- [13] Object Management Group (OMG) 2007. "UML 2.1.1, Superstructure Specification".
- [14] Nariman Mani, Vahid Garousi, Behrouz H. Far "Monitoring Multi-Agent System for Deadlock Detection Based on UML Models" 978-1-4244-1643-1/08, IEEE, 2008.
- [15] V.Gligor and S. Shattuck "Deadlock detection in distributed systems," IEEE Trans. Soft. Eng. pp. 435-440, 1980.
- [16] K. M. Chandy, J.Misra .and L. M. Haas." Distributed Deadlock Detection."ACM Trans. on Computer Systems, vol. 1, pp. 144-156, 1983.
- [17] OMG 2005. UML Profile for schedulibility, Performance and Time Specification. Adopted Specification <http://www.omg.org/docs/formal/05-01-02.pdf>

9. AUTHOR'S BIOGRAPHY

Dr. Gufran Ahamd Ansari received his Bachelor degree (B.Sc. Computer Science) from Shia P.G. College, Lucknow in 1997, Post graduate diploma from NIIT Lucknow, MCA from DR. B.R. Ambedkar University Agra in 2002 and Ph.D(Computer Science) from Babasaheb Bhimrao Ambedkar (A central) University, Lucknow, U.P., India in 2009. He is currently working as an Assistant Professor Department of Information Technology, Qassim University, Saudi Arabia. Earlier he worked as Associate Professor and head Department of Computer Science and Engineering at MIT Meerut. lecturer at Azad Institute of Engineering & Technology (AIET) Lucknow, Lecturer, Senior Lecturer and Assistant Professor at Institute of Foreign Trade & Management (I.F.T.M), Moradabad U.P., India. He has over 10 years of experience in teaching undergraduate as well as postgraduate students of Computer Science and Computer Applications. Currently he is actively engaged in the research work on domain based of real-time system modeling through the Unified Modeling Language (UML). He has produced several outstanding publications in National & International Journal on various research problems related to the domain based UML modeling & Security, Testing and Designing.