Aspect oriented databases

A Review Paper Writing Proposal

Submitted by

Anjali Kumari(11408495)

To

School of Computer Applications

In partial fulfillment of the Requirement for the

Award of the Degree of

Master of Computer Applications

Under the guidance of

Mr. Kumar Vishal

Certificate

This is to certify that **Anjali kumari(11408495)** have completed their MCA Research Paper Writing Proposal titled **“Aspect orientation on relational databases”** under my guidance and supervision. To the best of my knowledge, the present work is the result of their original investigation and study. No part of the dissertation proposal has ever been submitted to any other degree or diploma.

The proposal is fit for the submission and the partial fulfillment of the conditions for the award of the degree of Master in Computer Applications.

Date: ---

Signature of the Advisor

Name:

Anjali kumari

Aspect Orientation On Relational Databases

Abstract: *Aspect oriented programming has been developing at a rapid rate over the past few years the AOP programming has paved a way to reintroduce the programming paradigm completely into a new light the use of AOP is increasing over multiple researchers in the domain of the software engineering, this paper tries to bring forward all the benefits and the solidarity of the Aspect oriented programming language as a language of the future some concerns that are pointed out in this paper and the integration of OODB on to the framework is brought into the light.*

Keywords:*AOP,OODB,Aspect oriented programming.*

Acknowledgement

It is great pleasure to acknowledge the assistance of contribution of many individuals to this effort. We take this opportunity to thank all the people who have helped us through the course and producing this report. I sincerely thank Mr Kumar vishal for her guidance, help and motivation. Apart from the subject of our course, we learnt a lot from her, which we sure, will be useful in different stages of our life.

We are grateful to Mr. Rishi Chopra, Discipline of Computer Application for helping us in pursuing this topic in a smooth and organized manner.

We appreciate the contribution our friends whose name has not been listed here but has been much helpful not on the course of this project but also in the course of my study.

Lastly, but not least, we gratefully acknowledge the support, encouragement and patience of our families.

Anjali kumari

INTRODUCTION

In the past decade the Aspect oriented programming has drawn the interest of many researchers and computer programmers in the domain of the software engineering. The major role is played by the code repositories in order to recycle and mine the exiting assets available to us. Therefore its of the utmost importance that the new programming languages must have support for the existing database systems such programming languages must be able to extend the predefined structures of the wide databases being in the generation where big data plays a major role in the world wide web the applications implemented must e well compatible with them. This paper studies the existing relational databases and mapping the aspect orientation anatomy into the existing relational database models prior to the formation of queries that are fine grained in nature. This will eventually lead to the the much anticipated flexibility into the search and retrieval of data, this must be in contrast to the existing code repositories which stores the code on the memory as BLOBs which is complemented by the meta-data of the code.

The main aim is to make software development easier by introducing the support for modularity, aspects are abstraction that generally refers to solve various cross cutting problems that is the code cannot be confined to a single class but is fragmented into multiple module and then is distributed among the multiple classes, examples of aspect oriented architecture includes fault tolerance, management of the memory and communication of the data over multiple classes ,over coming real-time constraints, collective resource sharing, optimization of data performance,synchoronization and debugging. An aspect weaver is used to merge the multiple classes that are created and coded separately from the aspects.

The development phase under which th aspect oriented programming is being developed in is much identical to what object oriented programming paradigm went through in year 1960-70,as we see now the object oriented programming is being used widely across multiple domains of programming and software development activities such as design, creating models, analysis etc,moreover the databases are designed in such a way that it complements the object oriented structure, similar propagation can be seen in aspect orientation where its moving towards domains of specification and design,and the databases are however a territory unexplored. The persistence is however considered to be a part of the aspect orientation of the system aspects and the persistence in the database is however completely ignored until now.

2 Aspect oriented programming:

Aspect oriented programming works towards separation of the concerns by localizing cross cutting features such as code implementation in the memory management fault tolerance debugging the code and synchronizing amount the multiple classes that share the information among each other the code used for the debugging is distributed among all the classes whose instances must be debugged and synchronized ,such patterns are employed in order to deal with the various cross cutting concerns as the provide the guidelines or say hints of sort to implement a good structure which may not be suitable for all the classes and also might not provide a complete solution to the problem of code tangling. the classes are linked to aspects using the implicit and explicit join points the classes are coded separately from the codes the cross cut them. Theses cross cutting code is encapsulated by the AOP into separate constructs known as Aspects.

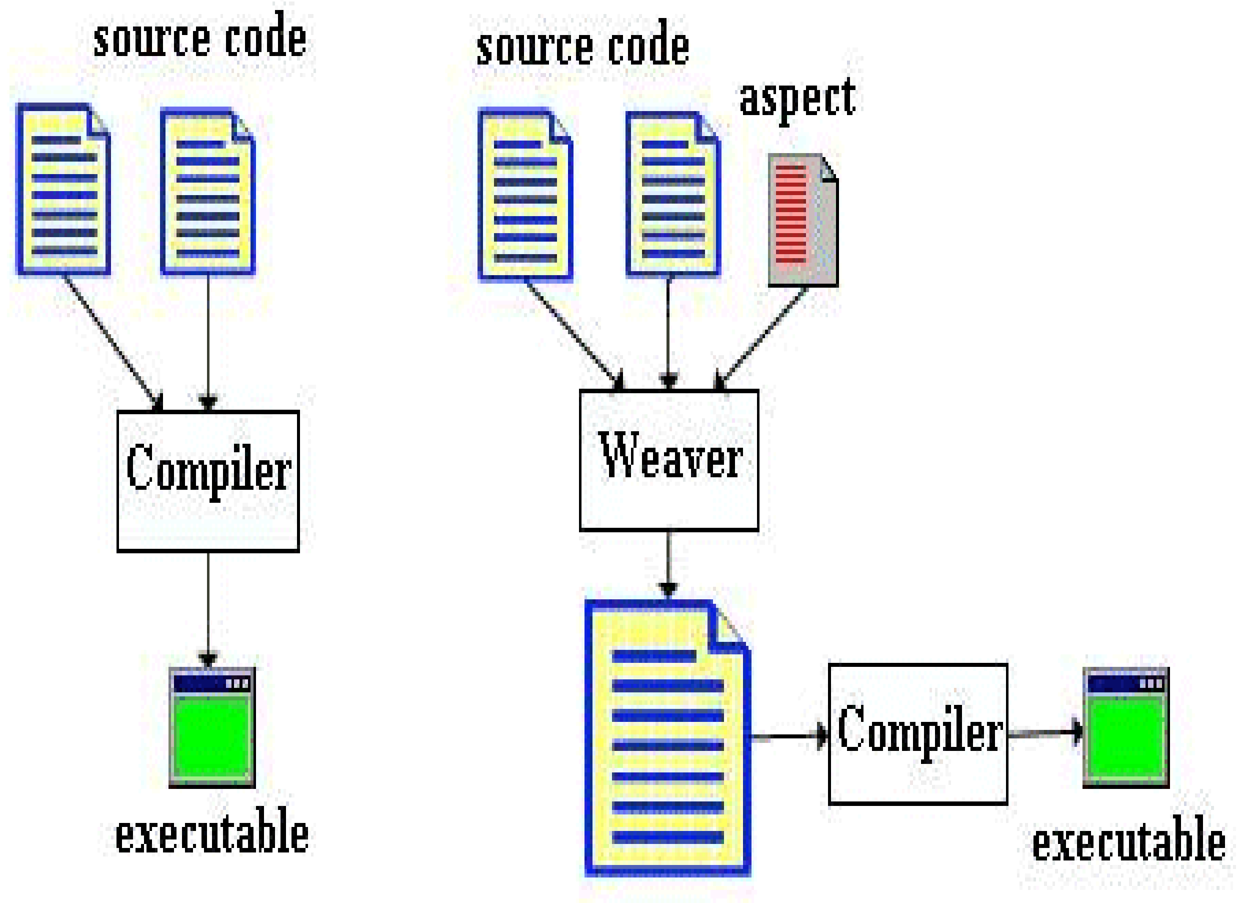
Categories:

- Open: the classes and the aspects are familiar to each other.
- Aspect-directional: the class is familiar with the aspect but it doesn't goes the other way round
- Object-directional: the aspect is familiar with the class but class isn't.
- Closed: neither aspect nor the class is familiar with each other.

Aspect Weaver:

The join points are used by the aspect weaver in order to merge the aspects and classes with each other. This can be done statically in form of the phase at the compile time or it can be done dynamically during the run time. various research and new AOP techniques are implemented in order improvise and improve the separation of the various concerns that exists in the AOP paradigm. Java uses an extension known as AspectJ its completely based on the AOP functionality, this extension enables us an environment to make aspect code from java class code. Similar extensions to other programming languages have also been developed Smalltalk an aspect oriented language of domain that deals in domain of robust programming is also found in aspect.other AOP approaches to achieve such similar results in separating the concerns are composition filters, subject-oriented programming and adaptivity programming. Subject oriented programming deals in the programming of different perspectives on the object model singular in the nature, various subject models are created in which the subject are nothing but the composition rule declared before hand.

CROSS-CUTTING CONCERN: Examples of crosscutting concerns would include exception handling, authentication and security aspects, which are considered important parts of many procedures or classes in conventional approaches. Therefore, they are handled in multiple locations within the same program, causing a drastic decrease in the quality, readability and modularity of the software.AOP treats Aspect differently. They are considered as an extended version of the class with additional features.



3 ASPECT ORIENTED DATABASE:

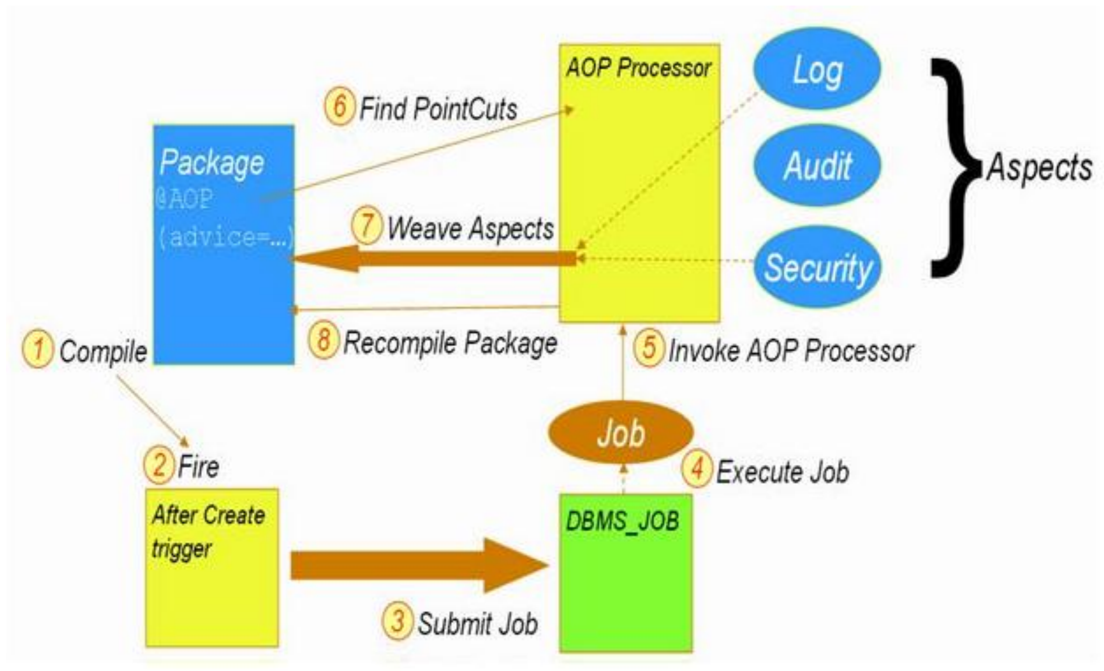
AOD enables the database to deal and store cross cutting data concerns a cross-cutting data concern is a data need that is globally used (potentially applicable to the entire database) and widespread (can be used to enhance many, different databases). Data has a wide variety of cross-cutting data concerns, including provenance, time, lineage, and security. So this research has the potential to impact every database. But developers currently have to rely on ad-hoc techniques to add these concerns to a data collection. To make the task easier, more flexible, and more general, we plan to apply techniques adapted from AOP to database management systems preserving three key benefits of the aspect-oriented approach.

Aspect independence-Aspects are designed, developed, and coded independent of a particular database.

Late binding –aspects can be woven into existing database.

Lightweight footprint – aspect weaving can be made easier using this.

in this approach the aspect is "tag" with metadata which in result creates AOD.when its attached the aspect becomes active whenever the connected data is used. in order to make this more feasible there is a need that data weaver should be extensible so that the developers may plug-in the semantics of different cross cutting data concerns this can be done by specifying only a few core functions, using this many different aspects can be supported in a parallel methodology.



The weaving process is implemented in java is done at three levels: compile time, class load time and runtime. In PL/SQL ,the concept of class load time and runtime redefinition is not feasible. So the option to implement the above concept is implementation of weaving during compile time. But weaving during the compile time is not quite flexible as the aspect change their exists a need to recompile again but this method is quite superior when seen from the performance point of view.

in PL/SQL the compilation phase occurs when the AFTER CREATE event takes place. This feature that was implemented in the oracle 9i enabled the trigger when the database is about to compile or just done compiling a given package, it can enable a trigger code that allows the user to react to the compilation process and the source code involved into it, using this we can weave the cross cutting concern into the metadata and recompile again.

in order to remove the interwoven aspects from the program there is a need of an easier way to do maintenance of the core code use of the EXECUTE IMMEDIATE statement just after the AFTER CREATE statement will lead to an error ORA-30511 as its an illegal operation,to bypass this problem we need to submit the job from the trigger itself that will eventually call the AOP_PROCESSOR package that will recompile the package after weaving the aspects that apply. This process will lead to delay of few seconds from te time the developer has notified the package is compiled and the moment the AOP process has done its work, its generally best to enable the AOP processor when the core concern is completed by the developer.

CONCLUSION:

As per this paper aspect can manage big databases easily because of its different aspects . Cross cutting concern is one of them. To make the task easier, more flexible, and more general, we plan to apply techniques adapted from AOP to database management systems preserving three key benefits of the aspect-oriented approach. Aspect get attach with metadata through metadata (data about data)operation performs on databases.

Refrences:

- 1>Aspect-Oriented Programming By:Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda,Cristina Lopes.
- 2>Design rule for Aspect Oriented ProgrammingBy:Rodrigo Bonifáciob ,Márcio Ribeirob,Carlos Eduardo ,Paulo Borbab
- 3>Design rule for Aspect Oriented ProgrammingBy:Alberto Costa Neto,Fernando Castorb
- 4>AspectJ (Aspect Oriented Programming in JAVA)By:DemeterJ,
- 5> AspectJ (Aspect Oriented Programming in JAVA)By:DemeterJ,
- 6>Aspect oriented programming in javaBy: Heba A. Kurdi

  7>Aspect-Oriented Programming Radical Research in ModularityBy: Gregor Kiczales.
- 8>An evaluation of aspect-oriented programming for Java-based real-time systems development.By: Shiu Lun Tsang ; Dept. of Comput. Sci., Trinity Coll., Dublin ; Clarke, S. ; Baniassad, E
- 9> The Next Steps For Aspect-Oriented Programming Languages (in Java)By: Jim HuguninXerox Palo Alto Research Center.
- 10> Distribution of the Object Oriented Databases.

By: A. MUŞAN, Marian P. CRISTESCU, Daniel I. HUNYADI, Lucian Blaga University of Sibiu Marius POPA, C.S.I.E. Faculty, A.S.E. Bucharest