# 60-510 LITERATURE REVIEW AND SURVEY

WINTER 2007

## Semantic-Based Concurrency Control in Object-Oriented Databases

SUBMITTED TO

Dr. Richard Frost

SUBMITTED BY

Deepa Saha


School of Computer Science

University of Windsor

# ABSTRACT

Object-Oriented database systems have been viewed as a promising technology as they can manipulate complex objects as well as provide support for encapsulated object. Various researchers, by exploiting the rich semantics of the object-oriented data model, claim that a better performance in concurrency control is possible to achieve. This report presents a comprehensive survey on the research contributions made by the researchers together with the directions they provide for future work. Though the main concern of this report is to review the work done in the field of semantic concurrency control in object oriented databases, it covers the semantics based concurrency control mechanisms of other databases too.

**Keywords:** Concurrency control, Object-Oriented database, Locking protocol, Transaction semantics.

## ACKNOWLEDGEMENT

I have the honor to acknowledge the knowledge and support I received from Dr. Richard Frost throughout this survey. I convey my deep regards to my supervisor Dr. Morrissey for encouraging me and helping me unconditionally. I would also like my family and friends for being my strength.

# CONTENT

# INTRODUCTION

Object-oriented Databases came into existence as a potential technology that can deal with complex objects and provide support for encapsulated objects. A concurrency control mechanism regulates the synchronization of access to the database so that the consistency of database is maintained. Supporting concurrency control in an OODB is more difficult compared than in a relational database because of some issues; the semantics of methods, nested method invocation and referentially shared objects being some of them. In OODBMS, the methods represent the behavioral aspects of objects and exploiting the semantics of these methods can help in eradicate data contention problem. If the semantic consistency of the execution result does not affect the execution order, then those executions can be performed concurrently. This leads to the conclusion that using the object structure as well as semantics of methods can elevate the degree of concurrency control which in turn will make OOBDs more suitable for performance-critical applications.

This survey is mainly dedicated to the works done in the field of semantic concurrency control in Object-oriented databases. But, apart from that, it covers most of the prominent works done on semantic concurrency control in relational databases, distributed databases, Real-time databases, Real-time OODBS and Active Object-oriented databases. This is done to provide a broader view of semantic concurrency control in general and again the different approaches used for semantic concurrency control in different databases, could be applied to Object-oriented databases too. All the papers selected for the surveying has been published in the last twenty five years and deals with concurrency control mechanism and the annotated twenty papers talk about the concept of semantic knowledge of transactions to be employed for concurrency control. Ten papers among these twenty are journal papers, eight are conference papers, one is a technical report and the remaining is a PhD thesis. Again, in all of these papers, semantic concurrency control methods are discussed but on different sorts of databases: one paper out of these deal with distributed database, four of them with relational databases, eight papers are actually on object- oriented databases, four paper deal with real-time databases, two of the papers are on real-time Object-oriented

database and only one paper is on active OODBMS. One noticeable thing about these papers is that most of the papers on this survey topic were published until 2000 and in 2000, the maximum number of papers was published but after that there were no such significant paper on it. It seems that most of the work in this field has been done by Jun et al. It was observed that five papers: [Molina, 1983], [Farrag et al. 1989], [Badrinath et al. 1990], [Muth et al. 1993], [Agarwal et al. 1993] among the annotated papers are the most referred papers in this area of research.

## Why Semantic Concurrency Control?

OODBs are a collection of classes and instances of those classes. Both of them are referred to as objects. Users access objects by invoking methods or by executing transactions, each of which is a partially ordered set of method invocations on a class or an instance object. By reasoning about schedules, transaction results and consistency constraints, a concurrency control mechanism can be designed which could provide an improvement in concurrency. Object- oriented databases provide greater opportunities for supporting semantic based concurrency control over traditional databases. OODBs have the capability of providing more semantic information about the operations which can be exploited in semantic concurrency control. Again, OODB allows concurrency control specific to individual data objects. Not only that, having improved capability of handling constraints also helps OODB to attempt for a semantic concurrency control. Any kind of protocol used in OOBD for concurrency control needs to address the issues of higher level semantics, nested executions and dynamic conflicts. So it seems that OODBS can provide higher concurrency among methods using behavioral properties of methods as, if semantics of operations are taken into account then multiple operations that do not affect the same part of an object can potentially execute concurrently. But this improvement of concurrency level seems possible only with the price of simplicity.

## Background

The problem of coordinating concurrent access to a database has been studied by many researchers. *Database System Concepts* [Silberschatz et al. 2006] can

be a good reference book for the initial concepts in this area. A number of research efforts have discussed the semantics based concurrency control concept and in the last few years this concept has been tried to be applicable for different sorts of databases. [Garcia-Molina, 1983] was the one proposing a scheme for transaction processing by exploiting semantic knowledge of transactions for distributed databases. The main idea that he pursued was to allow non-serializable schedules which has the capacity of preserving consistency as well acceptable to system users.

In relational databases this idea has been tried to implement. [Farrag – Ozsu, 1989] also used the same idea used by Garcia-Molina, but the only difference in their method was that they utilized different kinds of locks instead of the one used by Garcia-Molina for allowing non-serializable schedule and imposes fewer restrictions on the interleavings. [Badrinath et al. 1990] came up with a new multi-level concurrency control protocol which uses the concept of *recoverability*, a semantics-based notion of conflict. This paper is also the first one to report a performance evaluation of multi-level concurrency control protocols. As a generalization of conflict serializability, [Agarwal et al. 1993] introduced three new correctness criteria in this arena: *consistency, orderability* and *strong orderability.* [Grestl, 1998] provided a method with a better performance than two phase locking. In his method the transactions were decomposed into steps according the semantic criteria and then schedules according to his proposed method called Assertional Concurrency Control (ACC).

In Object-oriented databases, semantic concurrency control approaches are of two kinds which are discussed by [Lam et al. 1991]. They have classified the semantic approaches for concurrency control into transaction approach and data approach and described relevant models for both the approaches in their technical report. In this kind of databases, [Muth et al. 1993] presented a new semantics based locking protocol and their main principle was that commutative method execution on the same object was not considered as a conflict. [Resende et al. 1994] also provided a semantic locking but with the capability of handling the semantics of complex objects, nested executions and dynamic conflicts resulting from referentially shared objects. [Murakami et al. 1996] took into account both real time schedulability and Object oriented serializability for semantic concurrency control in OODB and presented a new real time scheduling protocol for it. In place Semantic Multi Granularity Locking (ISMGL), proposed by [Kwon et al. 1998], exploits semantics to maximize concurrency degree as well as multigranularity locking

to minimize locking overhead. In OODB, concurrency control mechanism of capable of dealing with semantics of methods, nested method invocation and referentially shared object was introduced by [Jun-Gruenwald, 1998]. [Madria et al. 2000] brought in a version of linear hash structure algorithm for using in multi level transaction model and utilized the semantics of the linear hash operation to provide more concurrency. [Jun, 2000] revisited his work of [Jun-Gruenwald, 1998] and extended it by doing a performance study by comparing with ORION2 and Malta's.

Real-time database has a few additional conditions to satisfy while semantic concurrency control is applied. [DiPippo et al. 1993] initiated a semantic concurrency control technique that supports such conditions as data temporal consistency and data logical consistency. [Peng et al. 1996] introduced a new method for this purpose which can increase the responsiveness of transactions. PRED-DF (PREDeclaration and Data Flow analysis based concurrency control protocol), provided by [Munnich, 2000], is also a semantics based method which is locking based, generates serializable schedules but works on main memory. A Colored Petri Net (CPN) model was used by [Neto et al. 2001] for producing a method to analyze semantic concurrency control.

In Real-time Object-oriented Databases, [Lee et al. 1994] exercise a semantic concurrency method that uses the notion of affected-set of operations to determine the compatibility relation of operations. [DiPippo et al. 1997] established a concurrency control mechanism for RTOODB that supports both logical and temporal consistency and at the same time the bounded imprecision that arises from their trade-off.

[Kangsabanik et al. 2000] seems to be the only researchers till now who provided a concurrency control mechanism for open tested transactions in an Active Object Oriented Database. It uses the semantics of transactions to achieve controlled cooperation and concurrency among transactions.

## Structure of the survey

This survey mainly focuses on the semantic concurrency control (SCC) methods in Object-oriented Databases but spans through other kinds of databases and the research done on these databases for applying semantics based concurrency control. Owing to this reason, this survey contains six more sections. All these sections are presented according to the ascending order of

their first paper's publication date. Section 2 covers a paper that deals with semantic concurrency control in distributed databases. Relational database and semantic concurrency control in them is briefly discussed in section 3. Section 4, on the other hand, contains a more detailed description of Object-oriented databases and how semantics based concurrency control methods can be applicable to them. The next chapter, section 5 deals with the developments of semantic concurrency control in Real-Time database. Section 6 deals with one paper published on Active OODB's semantic concurrency control and the last chapter, section 7, provides some concluding comments.

# SCC IN DISTRIBUTED DATABASES

## Problem description

A distributed database is basically a collection of databases residing at different computers or nodes but allows an integrated access. Like centralized databases, interference amidst concurrently functioning "user requests" or system failures leads the distributed database to an inconsistent state. As a consequence of that, researchers tried to extend their works of consistency concepts of centralized systems to distributed databases too.

As a matter of fact, consistency is a concept no dependent on the location of data and a schedule, even being executed in parallel by the nodes, can be symbolized as the execution of the actions. So, the mechanisms of centralized database for guaranteeing atomic transactions and serializable schedules can also be applied in a distributed system. But centralized databases do not suffer from the problem of large communication delays like Distributed database. So, the mechanism applied in centralized database system cannot perform satisfactorily in distributed databases. This fact leads researchers to look for other transaction management methods suitable for this kind of databases.

## Some approaches to ensure SCC in Distributed DB

*Non serializable schedules to process transactions*

In a distributed system, guaranteeing atomic transactions and serializable schedules always leads to some problems, like increased transaction delays due to conflicts among transactions, increased transaction delays due to

failures and loss of node autonomy. [Molina, 1983] proposed a transaction procession mechanism that might be a partial solution to these problems. He came up with the idea of *semantically consistent* schedules as in some applications even if the serializability is not maintained but the consistency is, then it is considered to be satisfactory from the perspective of higher parallelism and better performance. But if semantic consistency constraint is to be used, then transaction processing system should get some help from system users in the form of transaction semantic types, a distribution of transactions into steps, compatibility sets and counter steps. So, the main idea is to allow nonserializable schedules along with serializable ones, but these non serializable schedules have to preserve consistency and acceptable to system users and the author claims to improve performance by this method at least in some cases.

## Summary

The only paper discussed in this section is one of the earliest papers in the field of semantic concurrency control in any kind of database and again one of the most referred papers too. The main contribution of this paper is presented once more but in a tabular form in table 2.1.

| Year | Paper | Major Contribution |
|---|---|---|
| 1983 | Using semantic knowledge for transaction processing in a distributed database [Gacia-Molina] | Allows nonserializable schedules in their proposed SCC |

**Table 2.1**

# SCC IN RELATIONAL DATABASES

## Problem definition and motivation

In a database, concurrency control mechanisms ensure the consistency of the database while allowing a set of transactions to execute concurrently. If only syntactic information about transactions is known, then *serializability* is considered to be the main correctness criterion for concurrency control. A concurrent execution of a set of transactions or a schedule is said to be *serializable* if it is equivalent to a serial execution of the transactions. But with the advent of complex information systems, reevaluation of serializability as

the sole correctness criterion has surfaced. The main reason for this is the fact that these applications have much richer semantics in the underlying data than just *read* and *write* operations. If the semantic information of transactions is available, then the notion of serializability can be weakened and that can provide a higher level of concurrency.

## Different research aspects of SCC in DB

Semantics-based concurrency control protocols can be classified into two groups based on whether they are based on the semantics of transactions or on the semantics of objects. [Lamport, 1976] is the first researcher to introduce the semantics based concept and [Gray, 1981] pointed out the limitations of traditional transaction model and showed cases which take reap advantage from a semantics-based transaction model. In accordance with this situation researchers in this field have proposed new correctness criterions by relaxing the notion of consistency, correctness or conflict with the assumption that is semantic information about transactions are available than that could help in improving the performance.

*A new specification of consistency*

[Farrag-Ozsu, 1989] provided a new formulization to an existing work of [Garcia, 1983] where the semantic information about transactions is exploited and assume fewer restrictions on the interleavings among transactions than that of Garcia's. But they provide a new approach of specifying consistency. In their approach, they describe the allowable interleavings among transactions that are known to be correct and then ensure that each schedule produced by the system is equivalent to a correct schedule. They think this approach to be more suitable for using the semantic knowledge available about the transactions. In their semantic concurrency control mechanism, the semantic information of transactions is basically transaction types, transaction steps and transaction *breakpoints*. They define a new class of schedules called relatively consistent (RC) schedules and propose a semantic concurrency control method that produces only RC schedules.

*Semantics-based notion of conflict*

Simple semantics like synchronization properties of operations and the structure of operations can be used to enhance concurrency in complex

information systems and this has been shown by [Badrinath et al. 1990]. They present a new multi-level concurrency control that uses a weaker notion of conflict than commutativity, called *recoverability*. They have also developed the notion of *relative conflict* which is based on the structure of complex operations and is used to avoid deadlocks. This paper is the first one to report performance evaluation of a multi-level concurrency control protocol and the result of it shows that their method provides substantial performance improvement compared to that of a single-level two-phase locking based concurrency control scheme or even to that of a multi-level concurrency control scheme based on the concept of commutativity.

*Semantics-based correctness criteria*

A model for database system where the semantics of a database system has been captured by [Agarwal et al. 1993] and they have done it through abstract data types and the notion of *interaction*. Through an interaction a user can state consistency assertions which capture the user's view of the database before an operation is executed. They have defined three correctness criteria, *consistency* being the first of them. *Consistency* is based on the users' specification of consistency assertions and thus enables users to specify the permissible interleavings explicitly. But as this criterion turned out to be inadequate for databases where integrity constraint are not specified explicitly, two other correctness criteria called *orderability* and *strong orderability* came into existence. Both of these are based on the notion of equivalence to a serial schedule but the notion concept of equivalence was made more permissive. The authors believe to exploit the semantic information of databases more, while maintain the basic correctness principle of *serial* schedule.

*Assertional Concurrency control (ACC)*

[Grestl, 1998], wrote a dissertation, in which he proposes a scheme where transactions can be decomposed into steps depending on semantic criteria. The scheduling of the decomposed transaction is done by using Assertional Concurrency Control (ACC), an extension to the traditional concurrency control. ACC allows the partitioning of transactions into atomic, isolated steps and the scheduling happens according to a user provided specification. Grestl also proved that ACC has the capacity of providing better performance than a standard two-phase locking but only where lock contention is considered as a system bottleneck. Apart from that, ACC guarantees a semantic recovery, called *compensation*, in case of failure.

## Summary

This section comprised of four papers which deals with semantic concurrency control in relational database systems. This section dealt with semantics based notion of correctness, consistency, conflict and a new concurrency control mechanism. Table 3.1 will provide a zest of the papers discussed in this section.

| Year | Paper | Major Contribution |
|---|---|---|
| 1989 | Using semantic knowledge of transactions to increase concurrency [Farrag et al.] | Proposes a CC that produces *Relatively Consistent* schedules |
| 1990 | Performance evaluation of semantics-based multilevel concurrency control protocols [Badrinath et al.] | The application of *recoverability* and development of *relative conflict* |
| 1993 | Consistency and orderability: semantics-based correctness criteria for databases [Agarwal et al.] | Introduces three new correctness criteria as a generalization of serializability: *consistency*, *Orderability*, *strong orderability.* |
| 1998 | Semantic Concurrency Control Recovery and Performance Profiling for Improving Response Time in Database systems [Grestl, D.] | A new method -*(ACC)* Assertional Concurrency Control |

**Table 3.1**

# SCC IN OBJECT-ORIENTED DATABASE

## Problem definition and motivation

Current database applications are increasingly being taken over by Object-oriented techniques and that leads to the revaluation of traditional concurrency control mechanisms which are used for more efficiency. When database functionality was combined with object-oriented concepts then Object-oriented databases became the ideal information repository that is shared by multiple users and, multiple applications on different platforms. Though many of the techniques of traditional concurrency control can be carried over to object oriented databases but the model for transactions supported in conventional database is not suitable for long-duration transactions. New semantics in object-oriented databases need to be used to improve concurrency and to express correctness criteria other than serializability. It is felt that, there

is still a lack of efficient concurrency control in OODBMS. Conventional approaches for transaction management do not work well on OODBMS because of the complex objects and complex transactions OODB deals with. To avoid the potential data contention, most of the researchers have looked into the object structure and operation semantic while designing an appropriate CC mechanism. That is why; the researchers started exploiting the rich semantics of object-oriented data model to achieve better performance for OODBMS.

## Different approaches for SCC in OODBMS

Various researchers have taken up either the transaction semantics or the data semantics for SCC in OODBMS; others have used different locking models, multigranularity models and even multi-level transaction models. Not only that, different papers on this fields different issues like nested method invocation and referentially shared objects. This section tries to capture the main idea behind all these papers, this kind of database's semantic concurrency control being our main concern.

*SCC in OODB: approaches and models used*

A technical report was published by [Lam et al. 1991] which summarized the use of application and data semantics to optimize concurrency. They have explained three models of transaction approach: first of which is the compatibility set approach, the second one is constraint-based approach. The compatibility set based approach [Garcia, 1983] is one of the earliest efforts of using transactions semantics, whereas the constraint-based approach [Korth et al. 1988] has the capability of handling nested transactions in OODB. The last of this kind [Nodine, 1990] makes use of patterns to express correctness constraints to allow higher concurrency. The other approach called data approach contains two models. The first of them redefines a transaction as a sequence of typed operations on objects and the second one uses a serial dependency concept.

*Semantics of methods invoked on encapsulated objects*

[Muth et al. 1993] presented a semantic-based concurrency control scheme for OODB is. In this paper, the conflict between lower level operations or methods was ignored due to the commutativity of methods invoked at the higher level in nested method execution. They have made a lock required on

an object if a method or operation is invoked on the object. These locks are converted to retained locks at the end of a subtransaction. If a top-level transaction commits, all the locks held are released. They use semantics of methods so that, assuming two atomic operations conflict with each other, if they have ancestor which are compatible with each other and the ancestor of the lock holder commits, the lock request is granted. So, the lock request is never delayed until the top-level transaction commits. This can achieve higher concurrency. They have suggested the same principle when two methods conflict each other.

*Nested executions, dynamic conflicts*

A semantic two-phase locking protocol for OODB is presented in [Resende et al.1994]. They consider RSOs and nested method invocations in their work. They also exploit semantics of method for better concurrency. They have allowed any two methods to commute with each other if application programmers think their execution orders not to be important and this is done by using semantics of methods. Thus, by taking semantics into considerations, higher concurrency can be achieved. The only condition for this to work was that, the semantically commuting methods should be executed atomically. Again, in their work, locks are required only for atomic operations. The one drawback of this method is the big overhead which incurs while using locking for each atomic operation. This can lead to performance degradation in the presence of long-lived transactions.

*Enabling real-time schedulability in SCC of OODB*

[Murakami, 1996] initiated a method and an algorithm which is capable of real-time scheduling of the SCC in OODB. They describe their work as the realization of scheduling of concurrency control based on combination of database and operating system. Their new scheduling protocol considers object-oriented serializability as well as real-time schedulability. They mainly introduced the priority of process scheduling in the operating system. To avoid the worst case, they have made the early invoked transaction wait for long time to avoid deadlock in the scheduling of the Object-oriented Database systems. They have just proposed this approach but did not implement it.

*Enhanced Semantic Locking(ESL)*

[Kwon-moon, 1998] presented a locking scheme same as the locking scheme of [Resende et al. 1994] as they also consider semantics of methods and RSO.

In their method, called Enhanced Semantic Locking, they argue semantics can be provided at the discretion of the application programmer in methods. On the other hand, in order for ESL to support RSO, they adopt “in-place” conflict resolution policy. According to this, lock modes are not associated with methods. Commutativity of methods is determined when methods invoke shared subobject at the same time. This scheme requests a lock whenever a read or write atomic operation is invoked. But, ESL is different from [Resende et al. 1994] in that lock conversion for retained lock is prohibited. Like [Resende et al. 1994], it may incur big overhead. Also, an entire instance object locking granularity is adopted so that two methods accessing a disjoint set of attributes may conflict with each other.

*Reducing locking overhead and deadlocks*

[Jun. W et al. 1998] presented a locking-based concurrency control. It deals with three important issues in object-oriented databases: semantics of methods, nested method invocation and referentially shared object. In their proposed scheme, locks are required for the execution of each of the methods instead of atomic operations. This can reduce the locking overhead and possible deadlocks due to lock escalation. Also, a way of automating commutativity of methods is provided. In addition, concurrency is increased further by use of run-time information which can help in overcoming the problem of less concurrency due to locks by method invocations. One shortcoming of this method is that, they have not considered inheritance hierarchy.

*Use of multi-level transaction model*

[Madria et al. 2000] exploited semantics of multi-level transactions in the environment of linear hash structures to increase concurrency. They have designed a three-tier client/server layered architecture and an object-oriented implementation of multi-level transactions accessing linear hash structures. Multi-level transactions enhance the concurrency in the linear hash structure and handle transaction aborts. In this method, pages are modeled as objects whereas linear hash operations find, insert, delete, split, and merge are considered as methods. These methods correspond to multilevel transactions in their behavior and are implemented as threads and multithreads. The performance evaluation of the algorithm on the basis of the potential increase in concurrency and measuring the overheads associated with this system was not done.

*Semantics of methods, nested method invocation and RSO*

This work by [Jun, 2000] is an extension of [Jun et al. 1998] in which a locking-based concurrency control scheme in OODBs was presented. That scheme deals with three important issues in OODB concurrency control: semantics of methods, nested method invocations, and referentially shared object. In order to overcome low concurrency due to locks at the method level, run-time information is utilized to increase concurrency. In this paper, through simulation, the performance evaluation of that scheme is done with two existing works: Orion and Malta's and the results are analyzed. The performance study shows that the proposed scheme is superior to existing works. Though this scheme aims to stable OODB systems, the author claims to deal with evolving OODB systems as well as distributed OODBs in future.

## Summary

This section covered the eight important papers that contributed to the field of semantic concurrency related method and other related issues of SCC in OODB. The list of papers discussed over this section along with their key contribution is listed in table 4.1.

| **Year** | **Paper** | **Major Contribution** |
|---|---|---|
| 1991 | Concurrency Control in Object-Oriented Databases [Lam et al.] | Discusses semantic approaches: transaction and data approach in OODB for CC |
| 1993 | Semantic concurrency control in object-oriented database systems [Muth et al.] | A new locking protocol where commutative method execution is not considered as a conflict |
| 1994 | Semantic locking in object-oriented database systems [Resende et al.] | SCC incorporating semantics of objects, nested executions and dynamic conflict |
| 1996 | Real-time scheduling for semantic concurrency control of object-oriented database systems [Murakami et al. 1996] | A new RT-scheduling protocol for SCC of OODB |
| 1998 | Semantic multigranularity locking and its performance in object-oriented database systems [Kwon et al.] | A new protocol- In-place Semantic Multi Granularity Locking (ISMGL) |
| 1998 | Semantic-Based Concurrency Control in Object-Oriented Databases [Jun et al.] | Deals with semantics of methods, nested method execution & RSO |

| Year | Paper | Major Contribution |
|---|---|---|
| 2000 | Multi-level transaction model for semantic concurrency control in linear hash structures [Madriaa et al.] | Exploits the semantics of the linear hash operation at each level of transaction nesting to allow more concurrency |
| 2000 | Semantic-based locking technique in object-oriented databases [Jun] | Extension of [Jun et al. 1998] and performance study comparing with ORION and Malta's. |

# SCC IN REAL-TIME DATABASES

## Problem Definition

Many real-time systems need the real-time databases for supporting the large volumes of time-constrained data which are used by time-constrained transactions. Traditional database systems provide concurrency control techniques that try to preserve data logical consistency and transaction logical consistency. But in real-time databases, two additional requirements are there: data temporal consistency as well as transaction temporal consistency. But, again, methods that support one for of consistency requirement are not often well-suited for others. Many traditional concurrency control mechanisms support only serializable schedules but they are ill-suited for RT databases as they are only concerned with the logical correctness of data and transaction but not temporal criteria. Moreover, allowing only serial or serializable schedules is too restrictive for real- time databases as they must schedule transactions to meet timing constraints. This led the researchers into finding new SCC methods for real-time databases too.

## Different approaches for providing SCC in RT databases

This section covers six papers that discuss semantic concurrency control for RT databases. Three of them propose new methods; one ensures data logical and temporal consistency as well as the tradeoff between them, the next one proposes a method which is based on priority driven conflict resolution, the third one is a mechanism suitable for main-memory real time databases, the forth paper uses the concept of epsilon serializability, the next is a SCC

method that increases transaction responsiveness, and the last one analyzes the SCC in real-time databases.

*Use of real-time object-oriented database model*

[DiPippo et al. 1993] came up with a mechanism for concurrency control which was based on a real-time object-oriented database model. In this model, each object has its own compatibility function and it has the capability of expressing the conditional compatibility of any two potential concurrent operations on the object. The conditions are on semantics of objects like, allowable imprecision, accompanied by present system state on the object. This compatibility function expresses the allowable concurrency. Each individual object uses semantic locking to enforce it. This technique can support data temporal consistency and data logical consistency as well as the tradeoff between these two.

*Use of priority based conflict resolution*

[Lee et al. 1994] present a new design of Real-time Databases depending on the object-oriented data model and focused on the techniques to improve concurrency of transactions executing on data objects through the semantic information of operations defined on objects as object-oriented data model provide greater opportunities for supporting semantics-based concurrency control. They followed the approach of the notion of affected-set of operations to determine operation compatibilities and used concurrency control algorithms enhanced by priority based conflict resolution schemes. This method can decrease the complexity of the compatibility relation construction process but cannot maneuver the tradeoff between logical and temporal consistency.

*Increasing responsiveness of transactions*

As real-time applications must have access to the correct and up-to-date information to respond to real-world event, [Peng et al. 1996] proposed a SCC protocol that can increase the responsiveness of transactions in a real-time system. Their protocol defines a method compatibility matrix (CM) for each data object. A transaction using these object issues a sequence of method calls which are examined by the objects. The authors introduce a term called “external consistency” which defines the requirement for the timely interaction between a database and the physical world. The authors claim that their method enhances external consistency and suggest to combine their

technique with the two phase locking protocol or optimistic protocol to employ the riches object semantics to meet external consistency.

*Use of semantic locking*

[DiPippo et al. 1997] designed a concurrency control technique that could support expression and enforcements of the trade-offs between logical and temporal consistency constraints for real-time object-oriented databases. They named their method as *semantic locking*, which they designed for soft real-time data management. So, it can preserve temporal consistency but cannot guarantee to meet timing constraints and that makes this method inappropriate for hard real-time data management. In their semantic locking technique, concurrency control is distributed to individual data objects. It can also specify accumulation and bounding of logical imprecision that results from the trade-off of logical consistency for temporal consistency.

*PRED-DF- A data flow based SCC for RT main-memory DB*

A new semantic concurrency control called PRED-DF has been proposed by [Munnich, 2000]. This protocol is intended for main-memory real-time databases systems. It uses predecralation, is locking based and generates serializable schedules. It also does advance analysis of the application's data flow in order to minimize blocking times. The method's behavior is predictable: it does advance deadline verification with classic algorithms, which is needed for the verification of real-time requirements. There are some advantages of using this mechanism such as low cost, high efficiency and less pessimistic deadline verification as it locks whole data set, short locking time and as predeclaration with large lock is most efficient for main-memory databases.

*Analysis of SCC for RTDB using Colored Petri Nets*

[Neto, 2001] presented a paper with the objective of introducing a method for analyzing real-time database applications' semantic concurrency control. For doing this, first a Colored Petri Net (CPN) is built and a finite set of real-time transactions are modeled in CPN. The analysis of this model allows the verification and possibly the correction of specifications of logical as well as timing constraints to guarantee the scheduling of the transactions and committing of all the transactions satisfying the defined deadlines. Basicially, the model is analyzed to confirm whether the compatibility functions defined for the semantic concurrency control were correctly defined or not.

## Summary

The six papers discussed in this section are presented, in a nutshell, in the following table which shows the major contribution provided by each of these papers.

| Year | Paper | Major Contribution |
|---|---|---|
| 1993 | Object-based semantic real-time concurrency control [DiPippo et al.] | A technique that support data temporal and logical consistency |
| 1994 | Semantic-based concurrency control for object-oriented database systems supporting real-time applications [Lee et al.] | A SCC method that gives precedence to more urgent operations through the use of priority driven conflict resolution |
| 1996 | A Semantic-Based Concurrency Control Protocol for Real-Time Transactions [Peng et al.] | A SCC method that increases the responsiveness of transactions |
| 1997 | Object-Based Semantic Real-Time Concurrency Control with Bounded Imprecision [DiPippo et al.] | A SCC method that supports logical and temporal consistency as well as the bounded imprecision that results from their trade-off |
| 2000 | PRED-DF - a data flow based semantic concurrency control protocol for real-time main-memory database systems [Munnich] | A main memory SCC method called PREDeclaration and Data Flow based concurrency control |
| 2001 | Analysis of periodic transactions and semantic concurrency control for real-time databases using colored Petri nets [Neto et al.] | Introduces a method to analyze SCC using CPN |

**Table 5.1**

# SCC IN ACTIVE OBJECT-ORIENTED DATABASES

## Problem definition

In AODBMS, transactions can monitor the database objects and can take specific actions when a particular event occurs in the database objects. These are specified by Event-Condition-Action (ECA) rules. AODBMS has the capacity to handle long duration transaction and the transactions are viewed as

a collection of subtransactions. It is thought that AODBMS with ECA rules can capture the semantics of many real-life applications which are of long duration as well as cooperative. But ECA rule has no concept of correctness when the inter-transactional events and detached mode rules are taken into account. Again, the traditional notion of serializability is too restrictive a correctness criterion for long duration and cooperative transactions. To resolve these issues, research has been in this area and the only major paper found in this area has been discussed in this section.

## One approach of providing SCC in AODBMS

*NP-QuadLock in AODBMS*

[Kangsabanik et al. 2000] proposed a concurrency control scheme named NP-QuadLock for cooperative and long duration transaction management in AODBMS on the basis of an open nested transaction framework. It exploits the semantics of the transactions to achieve better cooperation and concurrency among the transactions. Atomic AODBMS transactions are treated as base transactions. A complex transaction type is formed from a collection of base and complex transactions, a set of detached mode ECA rules and a state transition model. The cooperation semantics of a complex transaction type with other complex transaction types is specified by associating with each state of a complex transaction, a set of cooperating complex transaction types. A correct class of schedule is produced. It is called CoopComp-schedule and it can satisfy the state transition semantics of the individual complex transactions and maintain the cooperation and compensation semantics of the individual complex transactions within the generated schedule.

## Summary

The only paper described in this section has been given in a tabular form in table 6.1 to specify its main contribution.

| **Year** | **Paper** | **Major contribution** |
|---|---|---|
| 2000 | Semantic-based concurrency control for object-oriented database systems supporting real-time applications [Kangsabanik et al.] | A SCC for open nested transactions to achieve controlled cooperation and concurrency. |

**Table 6.1**

# CONCLUSION AND FUTURE WORK

The objective of this survey is to review the major research work done in the area of semantic concurrency control in Object-Oriented Databases. While doing this, this report also covered all the significant work done in semantic concurrency control in other databases like, relational database, distributed database, real-time database and Active Object-oriented databases too. This article talks about the necessity of semantic concurrency control and the benefit of it in Object-oriented databases and also various issues related to this and discusses the relevant methods and summarizes the main concept behind all the work. But despite this, many researchers mention of resolving various issues and some more future work in this area.

[Muth et al. 1993] exploited the semantics of methods in their SCC mechanism but did not address the problem of non-disjoint complex objects. They faced with a problem with this kind of objects was the test of conflicts between method invocations on different but potentially non-disjoint complex objects. This issue still needs to be resolved. Again, this approach needs to be extended for recovery of multi-level transactions too.

[Murakami et al. 1996] presented a new real-time scheduling protocol for semantic concurrency control of object-oriented database systems but they have mentioned the implementation of their work as a future work.

[Jun et al. 1998] provided a locking-based concurrency control scheme in OODBMS but hey have not considered inheritance hierarchy where a subclass inherits or redefines super class of the subclass. So, that also remains to be an open area.

[Madria et al. 2000] introduced a multi-level transaction model for SCC, in which they refer the performance evaluation of their algorithm to find out the potential increase in concurrency and overhead measurement of their system as a future work.

[Jun, 2000] also came up with a technique for concurrency control but it was for stable OODBs only. But he mentioned the extension of his work to an evolving database, whose schema is constantly changing, as a future work.

From all these discussions, it seems there is still scope for improvements and extensions of the existing works of SCC in OODBMS. Again the concepts used for SCC in other databases can also be of help in OODBMS too.

# BIBLIOGRAPHY

AGRAWAL, D., ABBADI, A. E. AND SINGH A. K. 1993. Consistency and orderability: semantics-based correctness criteria for databases. *ACM Transactions on Database Systems (TODS), 18,* 3, 460-486.

AGRAWAL,D. AND ABBADI, A. E. 1994. A nonrestrictive concurrency control protocol for object-oriented databases. *Distributed and Parallel Databases 2, Kluwer Academic Publishers, Boston,* 7-31.

AGRAWAL, D., BRUNO,J. L., ABBADI, A. E. AND KRISHNASWAMY, V. 1994. Relative serializability: an approach for relaxing the atomicity of transactions. In *Proceedings of the Thirteenth ACM SIGACT-SIGMOD symposium on Principles of Database Systems,* 139-149.

AKINTOLA, A. A., ADEROUNMU, G.A. AND OSAKWE A.U. 2005. Performance Modeling of an Enhanced Optimistic Locking Architecture for Concurrency Control in a Distributed database systems. *Journal of Research and Practice in Information Technology, 37,* 4, 365-380.

ARUMUGAM, G. AND THANGARAJ, M. 2005. An efficient locking model for concurrency control in OODBS. *Data Science Journal, 4,* 59-66.

BADRINATH, B. R. AND RAMAMRITHAM, K. 1992. Semantics-based concurrency control beyond commutativity. *ACM Transactions on Database Systems, 17,* 1, 163-199.

BADRINATH, B. R. AND RAMAMRITHAM, K. 1990. Performance evaluation of semantics-based multilevel concurrency control protocols. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data, 163-172.*

CHANG, C .T. K. 2002. Adaptive multi-granularity locking protocol in object-oriented databases. *PhD thesis, University of Illinois, Urbana- Champaign.*

CHANG, Y. I., WU, C. C., ANN, H. Y. 1998. Hybrid approach to concurrency control for object-oriented database systems. In *Proceedings of NSC-Part A: Physical Science and Engineering (EI), 22,* 1, 78-94.

DIPIPPO, L. B. C. AND WOLFE, V.F. 1993. Object-based semantic real-time concurrency control. In *Proceedings of Real-Time Systems Symposium,* 87-96.

DIPIPPO, L. C. 1995. Semantic Real-Time Object-based Concurrency Control. *PhD thesis dissertation, University of Rhode Island.*

DIPIPPO, L. C. AND WOLFE, V. F. 1997. Object-Based Semantic Real-Time Concurrency Control with Bounded Imprecision. *IEEE Transactions on Knowledge and Data Engineering, 9,* 1, 135 - 147.

DIPIPPO, L. C. AND WOLFE, V. F. 1996. Performance of Object-Based Semantic Real-Time Concurrency. *Technical Report,University Of Rhode Island.*

DIPIPPO, L.C. AND WOLFE, V. F. 1996. Performance of Object-Based Semantic Real-Time Concurrency. *Technical Report,University Of Rhode Island.*

FARRAG, A. A. AND ÖZSU, M. T. 1989.Using semantic knowledge of transactions to increase concurrency. *ACM Transactions on Database Systems (TODS), 14,* 4, 503 - 525.

GRESTL, D. S. 1998. Semantic Concurrency Control Recovery and Performance Profiling for Improving Response Time in Database systems. *PhD thesis, State University of New york, Stony Brook.*

HADAEGH, A.R. 1997. Multiversion concurrency control in object-based systems. *PhD thesis, University of Manitoba, Winnipeg.*

JUN, W. 2000. Semantic-based locking technique in object-oriented databases. *Information and Software Technology, 42,* 8, 523-531.

JUN, W. 1999. A New Class Hierarchy Concurrency Control Technique in Object-Oriented Database Systems. In *prceedings of the Third East European Conference on Advances in Databases and Information Systems, ADBIS'99,*128-140.

JUN, W. 1999. Providing high concurrency in object-oriented database systems. In *proceedings of International Symposium on Database Applications in Non-Traditional Environments,* 403-406.

JUN, W. 1998. An integrated concurrency control in object-oriented database systems. *PhD Thesis, University of Oklahoma.*

JUN, W. AND GRUENWALD, L. 2000. An Optimal Locking Scheme in Object-Oriented Database Systems. In *Proceedings of the First International Conference on Web-Age Information Management,* 95 - 105.

JUN, W. AND GRUENWALD, L. 1998. An effective class hierarchy concurrency control technique in object-oriented database systems . *Information and Software Technology, 60,* 45-53.

JUN, W. AND GRUENWALD, L. 1998. Semantic-Based Concurrency Control in Object-Oriented Databases. *Journal of Object-Oriented Programming, 10,* 8, 33-39.

JUN, W. AND GRUENWALD, L. 1996. A Flexible Class Hierarchy Locking Technique in Object-Oriented Database System. In *proceedings of ISCA International Conference on Computers and Applications.*

JUN, W. AND GRUENWALD, L. 1995. Supporting Fine Concurrency in Real-Time Object-Oriented Databases. In *proceedings of the 1995 Arkansas Computer Conference,* 28-29.

JUN, W. AND KIM. K. 2000. A Revised Implicit Locking Scheme in Object-Oriented Database Systems. *High Performance Computign and Networking, LNCS,* 618-621.

KANGSABANIK, P., MALL, R. AND MAJUMDAR, A. K. 2000. Semantic Based Concurrency Control of Open Nested Transactions in Active Object Oriented Database Management Systems. *Distributed and Parallel Databases, 8,* 2, 181 - 222.

KUILBOER, J. P. G.1992. Concurrency control and performance analysis in an object-oriented database management system. *PhD thesis, University of Texas, Arlington.*

KWON,K. AND MOON, S. 1998. Semantic multigranularity locking and its performance in object-oriented database systems. *Journal of Systems Architecture, 44,* 12, 917-935.

LAM, W., WANG, Y. AND FENG, Y. 1991. Concurrency Control in Object-Oriented Databases. *Technical Report, University of Waterloo.*

LEE, J. AND SON, S. H. 1994. Semantic-based concurrency control for object-oriented database systems supporting real-time applications. In *Proceedings of the Sixth Euromicro Workshop on Real-Time Systems,* 156 - 161.

LEE, S. K., JUNG, S. Y. AND HWANG,C. S. 1996. A new conflict relation for concurrency control and recovery in object-based databases. In *Proceedings of the 5th International Conference on Information and Knowledfge Management,* 288-295.

LEE, S. Y. 1996. A Multi-Granularity Locking Model for Concurrency Control in Object-Oriented Database Systems. *IEEE Transactions on Knowledge and Data Engineering, 8,* 1, 144-156.

MADRIAA, S. K., TUBAISHATB, M. A. AND BHARGAVAA, B. 2000 . Multi-level transaction model for semantic concurrency control in linear hash structures. *Information and Software Technology Journal, Elsevier Science, 42,* 7, 445–464.

MALTA, C. AND MARTINEZ, J. 1995. Automating fine concurrency control in object-oriented databases. In *proceedings of International Conference on Computer Applications in Industry and Engineering,* 72-75.

MOLINA, H. G. 1983. Using semantic knowledge for transaction processing in a distributed database. *ACM Transactions on Database Systems (TODS), 8, 2, 186-213.*

MUNNICH, A. 2000. PRED-DF - a data flow based semantic concurrency control protocol for real-time main-memory database systems. In *Proceedings of the Seventh International Conference on Real-Time Systems and Applications (RTCSA'00),* 468-472.

MURAKAMI, Y., NISHIKAKU, M., OKADA, T. AND SAKAGUCHI, M. 1996. Real-time scheduling for semantic concurrency control of object-oriented database systems. In *Proceedings of the 7th International Workshop on Database and Expert Systems Applications,* 214-221.

MUTH, P., RAKOW, T. C., WEIKUM, G., BRÖSSLER, P. AND HASSE, C. 1993. Semantic concurrency control in object-oriented database systems. In *Proceedings of the Ninth International Conference on Data Engineering,* 233-242.

NETO, P. F. R., PERKUSICH, A., PERKUSICH, M. L. B. AND TURNELL, M. F.Q. V. 2001. Analysis of periodic transactions and semantic concurrency control for real-time databases using colored Petri nets. In *proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 4,* 2723-2728.

NETO, R., PERKUSICH,P. F., PERKUSICH, A., BARBOSA, M. L. 2003. Scheduling and Semantic Concurrency Control Analysis for real time databases. In *proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications, Nevada.*

PENG, C. S. 1998. Semantic-Based Concurrency Control protocol and distributed software agent for real time database systems. *PhD thesis, University of California, Irvine.*

PENG, C. S. AND LIN, K. J. 1996. A Semantic-Based Concurrency Control Protocol for Real-Time Transactions. In *proceedings of IEEE Real-Time Technology and Applications Symposium (RTAS '96),* 59-67.

POPOVICH, S.,WU, S. F. AND KAISER, G. E. 1991. An object-based approach to implementing distributed concurrency control. In *proceedings of the 11th International Conference on Distributed Computing Systems,* 65 – 72.

RESENDE, R. F., AGRAWAL, D. AND ABBADI, A. E. 1994. Semantic locking in object-oriented database systems. *ACM SIGPLAN Notices, 29,* 10, 388 – 402.

RUSO, D. 2001. A model for concurrency in object oriented database. *Technical report, Mcmaster university.*

SILBERSCHATZ, A., KORTH, H. AND SUDARSHAN, S. *Database System Concepts,* Fifth Edition, McGraw-Hill, 2006.

SHAH, P. AND WONG, J. 1996. Concurrency control in an object-oriented database system. *Journal of Systems and Software, 35,* 3, 169-183.

SHAPOSHNIKOV, A. 1998. Algorithms for efficient transaction management and consistent queries in client-server semantic object-oriented parallel databases. *PhD Thesis, Florida International University.*

WANG, S. 1990. Improvement of concurrency control within object-oriented database systems. In *Proceedings of the 1990 Symposium on Applied Computing,* 68-70.

WEI, H. C. 1993. Implementations of concurrency control and recovery for the object-oriented storage system. *M.Sc Thesis, University of Texas at Arlington.*

# APPENDIX

## Annotations of the main contributing papers of the field

GARCIA-MOLINA, H. 1983. Using semantic knowledge for transaction processing in a distributed database. *ACM Transactions on Database Systems (TODS), 8,* 2, 186-213.

This paper studies how transactions can be processed efficiently in a distributed database if the semantics of transactions are provided. The authors approached to create a general-purpose transaction mechanism system that will produce semantically consistent schedules and accepts “rules” that describe the most common features of semantically consistent schedules. For this, the transaction processing mechanism is provided with transaction semantic types, a transaction divided into steps, compatibility sets and counter steps, by the user. They believe that their method has the capability to improve performance.

The main theme applied by them is that the transactions to fall into a collection of semantic types and based on the semantic knowledge of the transactions and their actions, users can group these actions of the transactions into steps. This could enable them to asses which steps of a transaction of a given type can be interleaved without violating consistency with the steps of other transactions. Thus, this semantic knowledge can be provided to the transaction processing mechanism so that the non interfering transactions can be performed faster.

According to the authors, the proposed transaction processing mechanism is based on locking and the concept of locks are used in a novel fashion to make sure so that only compatible transactions are interleaved. The mechanism uses two kinds of locks: Local locks, used by each of the nodes in the system, ensure the execution of the steps is done as atomic actions and Global locks ensure that the interleavings of atomic steps do not violate consistency. The use of the combination of global and local locks made it possible for transactions to release their local locks after executing a revocable step and for compatible transactions to be interleaved. Again, the impact of blocked transactions is also minimized as here; it is possible for transactions, compatible with blocked transactions, to access locked objects.

Apart from these, the authors have tried to resolve some issues like deadlocks, transaction commit, multiple lock types, scheduling of lock requests and granularity of global locks that needs to be resolved before implementing the transaction processing mechanism. At the end, the authors claim that in applications where majority of the applications are local and most of them are compatible with the common non locals then using their method can improve performance.

- This work is related to:

GRAY, J.N. 1981. The transaction concept: virtues and limitations. In *Proc. Seventh Int. Conf. Very Large Data Buses (Cannes, France, Sept. S-11), ACM,* 144-154.

KORTH, H.F. 1981. Locking protocols: general lock classes and deadlock freedom. *Ph.D. dissertation, Dep. Electrical Engineering and Computer Science, Princeton Univ.*

- This work was later referred by:

AGRAWAL, D., ABBADI, A. E. AND SINGH, A. K. 1993. Consistency and orderability: semantics-based correctness criteria for databases. *ACM Transactions on Database Systems (TODS), 18,* 3, 460-486.

DIPIPPO, L.C. AND WOLFE, V.F. 1997. Object-Based Semantic Real-Time Concurrency Control with Bounded Imprecision. *IEEE Transactions on Knowledge and Data Engineering, 9,* 1, 135-147.

GRESTL, D. S. 1998. Semantic Concurrency Control Recovery and Performance Profiling for Improving Response Time in Database systems. *PhD thesis, State University of New York.*

KANGSABANIK, P., MALL, R., MAJUMDAR, A. K. 2000. Semantic Based Concurrency Control of Open Nested Transactions in Active Object Oriented Database Management Systems. *Distributed and Parallel Databases, 8, 2, 181-222.*

LAM, W., WANG, Y. AND FENG, Y. 1991. Concurrency Control in Object-Oriented Databases. *university of waterloo technical report.*

MUNNICH, A. 2000. PRED-DF - a data flow based semantic concurrency control protocol for real-time main-memory database systems. In *proceedings of the Seventh International Conference on Real-Time Systems and Applications (RTCSA'00),* 468-472.

MUTH P., RAKOW T. C., WEIKUM G., BRÖSSLER P. AND HASSE C. 1993. Semantic concurrency control in object-oriented database systems. In *Proceedings of the Ninth International Conference on Data Engineering,* 233-242.

PENG, C. S. AND LIN K. J. 1996. A Semantic-Based Concurrency Control Protocol for Real-Time Transactions. In *proceedings of the 2nd IEEE Real-Time Technology and Applications Symposium,* 59-67.

FARRAG, A.A. AND OZSU, M.T. 1989. Using Semantic Knowledge of Transactions to Increase Concurrency. *ACM Transactions on Database Systems (TODS), 14,* 4, 503-525.

This paper analyzes an existing approach of concurrency control which uses the semantic information of a transaction for allowing controlled nonserializable interleavings. According to the authors, this approach requires extra overhead while using the semantic information and it can only be helpful if the expense of serializable interleaving is too high. After examining the already existing method, they claim to establish a new formalization of it. The semantic information in the new formalization is decided to be transaction types, transaction steps and transaction breakpoints.

[Garcia-Molina 1983] first introduced the concept of exploiting semantic knowledge for concurrency control. Their concurrency control mechanism allowed both serializable and nonserializable schedules. In that method, the interleaving was specified with a concept of compatibility. The authors claim that their approach is more generalized as it imposes fewer restrictions on the allowable interleavings among the transactions than Garcia-Molina [1983] did and there are no hierarchical compatibility levels among transactions. The authors have defined consistency by ensuring that each schedule has the allowable interleaving among transactions and any schedule preserving consistency was named as a correct schedule. They believe that this approach helps in using the available semantic knowledge about transactions.

If the precedence graph of a schedule is acyclic and if ordering the nodes of that graph can yield into a correct schedule, than they have termed it to be Relatively Consistent (RC) schedule and proved that execution of correct schedules as well as RC schedules preserve consistency. The writers' concurrency control mechanism is a locking technique which uses various locks to allow nonserializable schedules and uses breakpoints to control interleavings. The authors declare that this produces only RC schedules and thus maintain consistency.

The authors admit that determining the correct interleavings is yet to be done and although this approach yields higher concurrency but gives more overhead. So, according to them, the tradeoff between these two things is an open research area.

- This work is related to:

GARCIA-MOLINA, H. 1983. Using semantic knowledge for transaction processing in a distributed database. *ACM Transactions on Database Systems (TODS), 8,* 2, 186-213.

LYNCH, N. 1983. Multilevel atomicity-A new correctness criterion for database concurrency control. *ACM Trans. Database Syst. 8, 4,* 484-502.

- This work was later referred by:

AGRAWAL, D., BRUNO, J. L., ABBADI, A. E. AND KRISHNASWAMY, V. 1994. Relative serializability: an approach for relaxing the atomicity of transactions. In *Proceedings of the Thirteenth ACM SIGACT-SIGMODSIGART symposium on Principles of Database Systems,* 139-149.

CHANG, Y. I., WU, C. C., ANN, H. Y. 1998. Hybrid approach to concurrency control for object-oriented database systems. In *Proceedings of NSC-Part A: Physical Science and Engineering (EI), 22, 1,* 78-94.

PENG, C. S. 1998. Semantic-Based Concurrency Control protocol and distributed software agent for real time database systems. *PhD thesis, University of California, Irvine.*

SHAH, P. AND WONG, J. 1996. Concurrency Control in an object-oriented database system. *Journal of Systems and Software, 35, 3, 169 - 183.*

BADRINATH, B. R. AND RAMAMRITHAM K. 1990. Performance evaluation of semantics-based multilevel concurrency control. In *Proceedings of the 1990 ACM SIGMOD international conference on Management of data,* 163-172.

This paper deals with concurrency control techniques that will be capable of handling high level abstract operations. As, according to the authors, presently available single level concurrency control mechanisms are unable to do so, they have introduced a new mechanism of multilevel concurrency control which is less restrictive and serves the high throughput demand. They have claimed their paper to be the first one to do a performance evaluation of multilevel concurrency protocols.

Their method of concurrency control was claimed to have used operation semantics, i.e., the synchronization properties of the operations as well as the hierarchical structure of operations. They demand their concurrency protocol uses a semantics-based notion of conflict which they named as recoverability, whereas traditionally other methods use conflict notions based on commutativity of operations. They explain their use of recoverability over commutativity by mentioning that though recoverability is weaker than commutativity but it provides more concurrency and at the same time retains the positive features of commutativity. The authors have applied the definition of relative conflict only to operations and sub-operations but not transactions.

In the new nested protocol of the authors, each operation is required to obtain a lock or a list according to the types of operation. It has rules similar to the nested two phase protocols but extends them by empowering them to track dependencies due to recoverability. A lock acquired for an operation can be released anytime but the list has to be empty to be released. They have claimed to guarantee the completion of operations in the same order of forming dependencies due to recoverability.

After doing a simulation study, the authors have reported that using semantics in multilevel concurrency control yields noticeable improvements in performance but the magnitude of it depends on both data and resource contention.

The authors have justified the complexity of protocol as well as the overhead in terms of maintaining commit dependency information by obtaining an overall improved performance.

- This work is related to:

BADRINATH, B. R. AND RAMAMNTHAM, K. 1987. Semantics based concurrency control Beyond Commutativity. In *Fourth IEEE Conference on Data Engineering,* 132-140.

BADRINATH, B. R. AND RAMAMRITHAM, K. 1989. Semantics based concurrency control Beyond Commutativity. *Submitted to ACM transactions on Database Systems.*

- This work was later referred by:

JUN, W. 1999. Providing high concurrency in object-oriented database systems. In *proceedings of International Symposium on Database Applications in Non-Traditional Environments,* 403-406.

LEE, S. K., JUNG, S. Y. AND HWANG, C. S.1996. A new conflict relation for concurrency control and recovery in object-based databases. In *Proceedings of the 5th International Conference on Information and Knowledge Management,* 288-295.

RUSO, D. 2001. A model for concurrency in object oriented database. *Technical report, Mcmaster University.*

LAM, W., WANG, Y. AND FENG, Y. 1991. Concurrency Control in Object-Oriented Databases. *university of waterloo technical report.*

This report discusses about the new semantics used in object-oriented databases to improve concurrency and express other correctness criteria apart from serializability. Concurrency is optimized using application semantics and data semantics. The models for these approaches are also specified.

The authors of this report describe the semantics approach of concurrency control in databases by classifying them into two groups: transaction approach and the data approach. Models for transaction approach make use of concurrency properties of transaction by the semantics of the transaction and the data they manipulate. On the other hand, a concurrency property on abstract data types is defined depending on the semantics of the type and its operations.

The authors report that in the transaction approach, the interleaving of concurrent transactions is clearly constrained by specifications on transactions. This approach requires central control with respect to each o the groups of concurrent transactions and operation semantics are to be considered with the arrival of the each new transaction. One good thing about this approach is that, it is easier for the application semantics to be added to the concurrency control. The authors talk about three models that are suitable for long-duration transaction while discussing the transaction approach of concurrency control. The first model divides each transaction into atomic steps and the transactions are categorized into different groups according to their semantic information. The second model is a formal model of nested transactions approach for a CAD environment whereas the third model provides a similar nested model but with a power of specifying the concurrency control with augmented finite state automata.

On the other hand, in the data approach, irrespective of the semantics of the applications an object provides a uniform and concurrent behavior. The semantics of applications are added as new operations on data types. Not only that, addition of application semantics might cause violation of modularity and addition of operations to a type might cause schema changes in this model. Two models of data approach are discussed. The first model is an extension of traditional transaction model but it defines a transaction as a sequence of typed operations on objects and these objects are instances of shared abstract types. The second model uses a serial dependency relation for defining operation conflicts and satisfies hybrid atomicity.

- This work is related to:

A. SKARRA AND S. ZDONIK. 1989. Concurrency control and object-oriented databases. In *Research Directions in Object-Oriented Programming, pages 395-421. MIT Press, Cambridge Mass.*

A. H. NODINE, M. H. SKARRA AND S.B. ZDONIK. 1990. Synchronization and recovery in cooperative transactions. In *The Fourth International Workshop on Persistent Object Systems,* 329-342.

- This work was later referred by:

KUILBOER, J. P. G. 1992. Concurrency control and performance analysis in an object-oriented database management system. *PhD thesis, University of Texas, Arlington.*

MUTH P., RAKOW T. C., WEIKUM G., BRÖSSLER P. AND HASSE C. 1993. Semantic concurrency control in object-oriented database systems. In *Proceedings of the Ninth International Conference on Data Engineering,* 233-242.

This paper provides a new locking protocol for object-oriented database system that ensures semantic serializability. The authors argue that if performance-critical OODBMS applications use the conventional methods for transaction management then because of excessive concurrency control conflicts, response time degradations can occur. According to them, their protocol improves the possible concurrency to a great extent as commutative method executions on the same object are not considered as a conflict.

In this paper the authors assume the complex objects to be disjoint and demand that their locking protocol exploits the semantics of methods on encapsulated objects as well as has the capacity to deal with complex objects. The central idea of the paper is to deal with the dynamic method invocation hierarchy of an OODBMS transaction as an open nested transaction where method executions that invoke further methods correspond to sub transactions.

In their proposed model, they state to handle the problem of bypassing the encapsulation of objects in a reasonably efficient way. In their protocol, locks of committed sub transactions are preserved to determine all conflicts between OODBMS transactions even if encapsulated objects are bypassed. Apart from that, the ancestors of the conflicting actions are used to decide whether the actions have a commutative ancestor pair or not.

According to the authors' opinion, this protocol is an extended locking protocol for open nested transactions in OODBMSs. Not only they present the method as a beneficial one as its' exploitation of semantics of methods enhance concurrency and it does not impose any kind of restrictions in the way in which methods are invoked. Besides, it is claimed to preserve conventional page or record-oriented locking protocols as special cases. In the locking protocol, the authors have re-evaluated the commutativity of actions on different objects whenever there is a lock conflict on a potentially shared component during the execution of the actions' descendants.

The authors state that they have not addressed the problem of non-disjoint complex objects. Though they refer their algorithm to be able to ensure semantic serializability even in the presence of non-disjoint complex objects but only in principle and admit that there are a lot of implementation issues that still needs to be addressed.

- This work is related to:

WEIKUM, G., SCHEK, H.J., 1992. Concepts and Applications of Multilevel Transactions and Open Nested Transactions. *Database Transaction Models for Advanced Applications, Morgan Kaufmann.*

MOSS, J. E. B. 1985. Nested Transactions: An Approach to Reliable Distributed Computing, *MIT Press.*

- This work was later referred by:

CHANG, C .T. K. 2002. Adaptive multi-granularity locking protocol in object-oriented databases. *PhD thesis, University of Illinois, Urbana-Champaign.*

KWON, K. AND MOON, S.1998. Semantic multigranularity locking and its performance in object-oriented database systems. *Journal of systems Architecture, 44,* 12, 917-935.

LEE, J. AND SON, S. H. 1994. Semantic-based concurrency control for object-oriented database systems supporting real-time applications. In *Proceedings of the Sixth Euromicro Workshop on Real-Time Systems,* 156 - 161.

RESENDE, R. F., AGRAWAL, D. AND ABBADI, A. E.1994. Semantic locking in object-oriented database systems. *ACM SIGPLAN Notices, 29, 10,* 388-402.

DIPIPPO, L.B.C. AND WOLFE, V.F. 1993. Object-based semantic real-time concurrency control. In *Proceedings of Real-Time Systems Symposium,* 87-96.

This paper talks about the design of an object-based semantic concurrency control method that can meet the real-time concurrency control requirements, namely, data temporal consistency and data logical consistency. It is claimed that a concurrency control technique that can support all the requirements is difficult to design as they might possesses fundamental conflicts at times. So for the decision about the trade-off between them the authors have depended on the system conditions as well as on the application semantics.

The authors' modeled a real-time database and in that each of the objects is provided with a distinct compatibility function to grant semantic locks to transactions. The compatibility function is used to express the conditional compatibility of any two potential concurrent operations on the object whereas the semantic locks enable transactions to invoke specified methods of the object. Apart from maintaining these requirements, the paper discusses how to handle data consistency, both temporal and logical and makes a best effort to meet timing constraints but cannot guarantee it. They have assumed the database manager uses a real-time, priority-based, preemptive scheduling of execution on the processor and the transactions use a two-phase locking scheme to coordinate transaction logical correctness.

In their process, if a transaction requests a semantic lock from an object, then either a lock is granted immediately or the request is placed on a priority queue within the object depending on the evaluation result of the compatibility function of the object. Unlike other processes, the condition in the compatibility function is user-defined conditions which are based on the semantics of the application. The semantic lock is either released by request of the holding transaction or implicitly released upon completion of method execution or when a transaction commits or aborts. Whenever a lock is released, the priority queue is checked for any requests that might be granted.

The authors admit that they have considered only concurrency control mechanism within a single object but that is more tractable compared to having considered the semantics of all objects and all transactions. The authors also demand to be in the process of developing a semantic locking technique which will be propagating from object to object via relationship objects in order to enforce inter-object constraints.

- This work is related to:

B. BADRINATH AND K. RAMAMRITHAM. 1988. Synchronizing transactions on objects. *IEEE Transactions on Computers, 37,* 541-547.

V. F. WOLFE, S. DAVIDSON, AND I. LEE. 1993. RTC: language support for real-time concurrency. *Real- Time Systems, 5, 1, 63-87.*

- This work was later referred by:

NETO, P. F. R., PERKUSICH, A., PERKUSICH, M. L. B. AND TURNELL, M. F.Q. V. 2001. Analysis of periodic transactions and semantic concurrency control for real-time databases using colored Petri nets. In *proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 4,* 2723-2728.

PENG, C. S. 1998. Semantic-Based Concurrency Control protocol and distributed software agent for real time database systems. *PhD thesis, University of California, Irvine.*

LEE, J. AND SON, S. H. 1994. Semantic-based concurrency control for object-oriented database systems supporting real-time applications. In *Proceedings of the Sixth Euromicro Workshop on Real-Time Systems,* 156 - 161.

AGRAWAL, D., ABBADI, A. E. AND SINGH, A. K. 1993. Consistency and orderability: semantics-based correctness criteria for databases. *ACM Transactions on Database Systems (TODS), 18,* 3, 460-486.

This paper mainly examines the semantics of objects as well as the transactions in database systems. According to the authors, as semantic based concurrency control protocols encompass the methods based on the semantics of transaction and the methods based on the semantics of objects and by using both these concepts provide the benefits of both the approaches and thus increase the concurrency in a database system. So, they propose to work by exploiting semantics of user programs as well as the objects they operate on and by dividing the responsibility of correct executions between the user and the scheduler.

According to the authors, the model presented by them in this paper is an extension of the existing works. They provide the semantics of database systems through abstract data types and the structure of user transactions by the notion of an interaction. They use the concept of *consistency assertions*. This captures the user's view of database before an operation is executed and is used as a user-defined correctness criterion for concurrency of transactions. This correctness criterion is named as *consistency*. It gives the users the flexibility of analyzing the different interactions and specifying the permissible interleavings.

Despite of the simplicity of implementation of this concept, it seems to be inadequate for applications where database integrity constraints are not given explicitly and the databases for which the users want to see consistent values. To resolve these issues, the authors propose two new correctness criteria which include consistency assertions as well as abstract data type definitions, namely *orderability* and *strong orderability.* Orderablibility is a generalization of view serializability but it was found that deciding whether a history is orderable is NP-Complete process. So, the concept of strong orderability came in, which is an alternative to the previous correctness criteria and is implementable. This criterion is obtained by generalizing the conflict serializability in two ways, by including the idea of consistency assertions and

by using reductions based on one-way commutativity of operations. Both of these are based on a weak notion of equivalence to a serial schedule.

The authors claim that by weakening these requirements has enabled them to exploit the semantic information of databases and interactions but by maintain the basic correctness principle based on serial histories.

- This work is related to:

HADZILAGOS, T. AND HADZILACOS, V. 1991. Transaction synchronization in object bases. *Journal of Computer Systems and Science, 43*, 2-24.

KORTH, H. F. AND SPEEGLE, G. D.1988. Formal model of correctness without serializability. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 397-386.

- This work was later referred by:

JUN, W. AND GRUENWALD, L. 1998. Semantic-Based Concurrency Control in Object-Oriented Databases. *Journal of Object-Oriented Programming, 10, 8,* 33-39.

DIPIPPO, L. C. AND WOLFE, V. F. 1997. Object-Based Semantic Real-Time Concurrency Control with Bounded Imprecision. *IEEE Transactions on Knowledge and Data Engineering, 9,* 1, 135 - 147.

MALTA, C. AND MARTINEZ, J.1995. Automating fine concurrency control in object-oriented databases. *In proceedings of International Conference on Computer Applications in Industry and Engineering,* 72-75.

RESENDE, R.F., AGRAWAL D. AND ABBADI A. E. 1994. Semantic locking in object-oriented database systems. In *proceedings of the ninth annual conference on Object-oriented programming systems, language, and applications*, 388-402.

This paper has a novel semantic locking protocol that uses semantics to increase concurrency in object-oriented databases. This protocol addresses the issues of complex object semantics, nested executions and dynamic conflicts of referentially shared objects. According to the authors, their method serializes transactions at the level of method executions, which means that the protocol uses the semantics of method's execution. The authors felt the necessity to address this issue as referential sharing among unrelated objects is bound to happen because in OODB new objects are composed from existing objects. In particular, their method is suitable for referentially shared objects where the referential sharing is determined dynamically during the execution.

In their method, the authors have assumed the objects to be ordered in a hierarchy and a method in one object can only call methods on objects that are

lower in the hierarchy. Again, it was also presumed that the commutativity relationships are defined for atomic operations are well-defined and can be derived based on the semantics and the specifications of the class and its methods. The conflict and commutativity relations between methods that referentially share sub-objects will be determined dynamically during the execution of these methods. So, tests for conflicts are done only for methods executing at the same object and methods of different objects will be executed concurrently and based on their executions, this protocol will capture the conflict relations among these methods.

In the proposed model, semantic locking, like nested two phase locking, grants a lock when no other method execution retains a conflicting lock or when in the presence of conflicting locks, the ancestors are retaining them. Apart from these two cases, if there are no conflicting locks retained by non-ancestors then one of the ancestors of the operation up to but not including its retainer and some ancestor of the requestor commute and then also this method grants a semantic lock. In case of implementation, they admit to generalize the one of nested two-phase locking protocol.

As a concluding note, the authors emphasize the point that this protocol is dependent on the assumption than objects are designed modularly and are organized in a static manner. But if this restriction can be enforced dynamically then it will be ensured that no incorrect execution is there without enforcing a static strategy on the objects.

- This work is related to:

C. BEERI, P. A. BERNSTEIN, AND N. GOODMAN. 1989. A Model for Concurrency in Nested Transactions Systems. *Jounal of the ACM, 36, 2,* 230-269.

D. AGRAWAL AND A. EL ABBADI. 1994. A Nonrestrictive Concurrency Control Protocol for Object Oriented Databases. *Distributed and Parallel Databases, an International Journal, 2,* 1, 7-31.

T. HADZILACOS AND V. HADZILACOS. 1991. Transaction Synchronization in Object Bases. *Journal of Computer and System Sciences, 43,* 1, 2-24.

- This work was later referred by:

MUNNICH, A. 2000. PRED-DF - a data flow based semantic concurrency control protocol for real-time main-memory database systems. In *Proceedings of the Seventh International Conference on Real-Time Systems and Applications (RTCSA'00),* 468-472.

JUN, W. AND GRUENWALD, L. 1998. Semantic-Based Concurrency Control in Object-Oriented Databases. *Journal of Object-Oriented Programming, 10,* 8, 33-39.

KWON, K. AND MOON, S. 1998. Semantic multigranularity locking and its performance in object-oriented database systems. *Journal of Systems Architecture, 44,* 12, 917-935.

LEE, J. AND SON S.H. 1994. Semantic-based concurrency control for object-oriented database systems supporting real-time applications. In *Proceedings of the Sixth Euromicro Workshop on Real-Time Systems,* 156-161.

This paper proposes an approach to solve the major issues concerned in designing semantic-based concurrency control for object-oriented database systems supporting real-time applications. Their approach depends on the notion of affected-set of operations to determine operation compatibilities, uses concurrency control algorithms which are based on priority-based conflict resolution and improves concurrency of transactions executing on data objects through the use of semantic information of operations defined on objects. According to the authors' claim, their method can radically reduce the complexity of the compatibility relation construction.

In order to reduce the complexity of design, the authors have used only serializability as the correctness criterion for logical data consistency and one concurrency control protocol has been used for all the objects of the system. These two assumptions simplify inter - object inconsistency problem. The authors have not considered temporal consistency constraints while deciding compatibility relation of operations which has helped them in alleviating the compatibility complexity. While determining operation compatibility, the notion of an affected-set is used. The affected set of each operation is constructed from its semantic specification and commutativity has be taken up as a basis for determining whether a particular operation invocation can be allowed concurrently with those in progress.

In the authors' opinion, this proposed arrangement enables the object type designer to specify the semantics of operations only and their compatibilities are determined from these specifications systematically. The authors declare that they attempted to show that optimistic concurrency control can be adopted in semantic-based concurrency control mechanism. Not only that, they claim their approach to have a less complexity in determining operation compatibility and it can assure inter-object consistency with some overhead for keeping information of the present status of executing transactions. Apart from these, the authors admit that, though they could reduce complexity, they were unable to maneuver the trade-off between

logical and temporal constraints and the concept of commutativity was not fully exploited as the attributes directly accessed by object operations were only considered to detect compatibility relation. The authors wished to evaluate the performance of their proposed method as a future work.

- This work is related to:

DiPippo, L. B. C. and V. F. Wolfe. 1993. Object-Based Semantic Real-Time Concurrency Control. In *Proceedings of the 14th IEEE Real-Time System Symposium.*

Muth, P., T. C. Rakow, G. Weikum, P. Brossler, C. Hasse. 1993. Semantic Concurrency Control in Object-Oriented Database Systems. In *Proceedings of the 9th International Conference on Data Engineering.*

- This work was later referred by:

Lee, S. Y. 1996. A Multi-Granularity Locking Model for Concurrency Control in Object-Oriented Database Systems. *IEEE Transactions on Knowledge and Data Engineering, 8,* 1, 144-156.

Peng, C. S. and Lin, K. J. 1996. A Semantic-Based Concurrency Control Protocol for Real-Time Transactions. In *proceedings of IEEE Real-Time Technology and Applications Symposium (RTAS '96),* 59-67.

Peng, C. S. 1998. Semantic-Based Concurrency Control protocol and distributed software agent for real time database systems. *PhD thesis, University of California, Irvine.*

Peng, C. S. and Lin K. J. 1996. A Semantic-Based Concurrency Control Protocol for Real-Time Transactions. In *proceedings of the 2nd IEEE Real-Time Technology and Applications Symposium,* 59-67.

This paper offers a semantic-based concurrency control protocol which the authors claim to enhance external consistency. It is aimed for real-time applications where user transactions must always have the most recent data. Besides, they have combined their proposed method with the two-phase locking protocol and with the optimistic concurrency control protocol in order to compare the improvement their method can bring in.

Though, there are a couple of methods that exploits the semantics of real-time data objects, they suffer from high-complexity and overhead. Again, in some cases, the external consistency is more desirable or critical than serializability. Authors state that the protocol can enforce or relax the serializability property based on the transaction semantics. In the paper, external consistency means that the difference between the time when a transaction access an external data item and the time when the data value valid

should be less than a threshold. Maintaining external consistency might lead to sacrifice of internal consistency if a transaction is associated with a deadline constraint.

The authors used a compatibility matrix (CM), which is created for each object in the database. This CM enables employing richer object semantics to meet external consistency requirement. In OODB, each object has an internal state which the object abstraction protects. By defining compatibility matrix (CM), several concurrent executions of the same method have been allowed as long as the CM for the object accepts them. If an object has n defined methods then the CM would have n×n entries and CM (a,b) will be checked if an instance of method a is being executed while method b is requested by a transaction.

The performance of the protocol has been evaluated by simulation. The authors have compared the average age factor (data age divided by the update period of the data being accessed) of completed user transactions and the percentage of both user and external transactions missing their deadlines. They have used the basic two-phase locking protocol with pre-locking with semantic two-phase locking and optimistic concurrency control (OOC) with semantic OOC. The results show that external consistency for transactions is enhanced using semantic two-phase locking but for semantic OOC less user transactions are aborted which means external consistency degradation in user transactions.

- This work is related to:

LIN, K. J.1989. Consistency Issues in Real-Time Database Systems. In *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences,* 654-661.

SONG, X. AND LIU, J.1995. Maintaining Temporal Consistency: Pessimistic vs. Optimistic Concurrency Control. *IEEE Transactions on Knowledge and Data Engineering,* 787-796.

- This work was later referred by:

PENG, C. S. 1998. Semantic-Based Concurrency Control protocol and distributed software agent for real time database systems. *PhD thesis, University of California, Irvine.*

MURAKAMI, Y., NISHIKAKU M., OKADA T. AND SAKAGUCHI, M. 1996. Real-time scheduling for semantic concurrency control of object-oriented database systems. In *proceedings of the 7th International Workshop on Database and Expert Systems Applications,* 214-221.

This paper handles both object-oriented serializability and real-time scheduling as well as proposes a real-time scheduling for semantic concurrency control of object-oriented databases. According to the authors, object-oriented serializability is capable of semantic concurrency control but not efficient in case of real-time scheduling. So, in this paper a real-time scheduling for semantic concurrency control of object-oriented system in order to obtain schedulability and deadlock avoidance by presenting the concept of priority of process scheduling instead of retained lock of semantic concurrency control has been provided.

Muth et al. [1993] examined a semantic concurrency control in object-oriented database based on object-oriented serializability. In this protocol, the decision of executing a subtransaction is dependent on commutativity or conflict between subtransactions accessing the same object. Apart from this, once a subtransaction is committed, it is converted into retained lock till the transaction completes, the one to which the committed subtransaction belongs. Besides, in case of conflict, this protocol gives “the worst case” in which the requested subtransaction may wait till the commit of top-level transaction of the preceding subtransaction. This paper approaches to avoid this worst case scenario and deadlock to enable schedulability of OODB.

For providing real-time scheduling for semantic concurrency control between transactions or subtransactions in OODB, the authors propose to give priority to each of the transactions and this to be invoked in order of top level transaction. Along with this, the executing order between conflicting subtransactions will be according to the priorities. The executing order between commuting subtransactions are recommended to be decide using priority ceiling. The concept of priority ceiling is used due to the uncertain assignment of execution between commuting subtransactions. While more than one subtransaction blocks a subtransaction from executing, the subtransaction with highest priority ceiling will be made feasible.

The authors have claimed that their method holds object-oriented serializability as well as capable of avoiding “worst case” of semantic concurrency control and deadlock. They have mentioned the implementation of their method as the future work.

- This work is related to:

MUTH, P., RAKOW, T. C., WEIKUM, G., BROSSLER, P.AND HASSE, C. 1993. Semantic Concurrency Control in Object-Oriented Database Systems. In *Proceedings on IEEE 9th International Conference on Data Engineering,* 23-32.

SHA, L., RAJKUMAR, R. AND LEHOCZKY, J. P. 1990. Priority Inheritance Protocols : An Approach to Real-time Synchronization. *IEEE Transactions on Computers, 39, 9,* 1175-1185.

- This work was later referred by:

  No relevant papers were found.

DIPIPPO, L.C. AND WOLFE, V.F. 1997. Object-Based Semantic Real-Time Concurrency Control with Bounded Imprecision. *IEEE Transactions on Knowledge and Data Engineering, 9,* 1, 135-147.

The paper illustrates a model and an object-oriented semantic real-time concurrency control technique for databases and it is demanded to support both logical consistency and temporal consistency. Not only that, it has been demonstrated that it can bound the imprecision, which is introduced at the time of trade-off between these two consistencies.

DiPippo-Wolfe [1993] designed a concurrency control method called semantic locking which supports the enforcement of the trade-offs between logical and temporal consistency constraints for real-time OODB. In this method, concurrency control is distributed to the individual data objects, each of which controls access to itself based on compatibility function for the object's methods. The compatibility function over there was defined semantically.

In this paper of theirs, they describe how their locking method can specify accumulation and bounding of logical imprecision that results due to trading off logical consistency for temporal consistency. In this protocol, the user defines the allowed amount of imprecision for a given operation invocation. The protocol determines whether the operation can execute concurrently with the active operations with the help of a modified commutativity table. The authors derived two general restrictions on the expressed semantics and state that they are sufficient for bounding logical imprecision. They have used Epsilon serializability, a correctness criterion specifying the correctness of a schedule for transaction execution to be correct if the results of the schedule are within specified limits of a serializable schedule, to prove sufficiency of the restrictions and thus claim to provide logical correctness while better enforcing temporal consistency.

They admit that their technique has high complexity as well as additional overhead for the run-time system to grant locks. The performance measurement showed that it takes on the order of hundreds of microseconds to execute semantic locking. But according to the authors, this is not too unreasonable and it indicates that their method is not appropriate for applications with short method executions and lock durations. But, they demand that performance comparison of semantic locking with other lock-based methods shows that semantic locking is generally better at meeting timing constraints that other techniques tested by them.

- This work is related to:

DIPIPPO, L. B. C. AND WOLFE, V. F. 1993. Object-Based Semantic Real-Time Concurrency Control. In *Proceedings of IEEE Real-Time Systems Symposium.*

RAMAMRITHAM, K. AND PU, C. 1995. A Formal Characterization of Epsilon Serializability. *IEEE Transaction on Knowledge and Data Engineering. 7,* 6, 997-1007.

- This work was later referred by:

NETO, P. F. R., PERKUSICH, A., PERKUSICH, M. L. B. AND TURNELL, M. F.Q. V. 2001. Analysis of periodic transactions and semantic concurrency control for real-time databases using colored Petri nets. In *proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 4,* 2723-2728.

KWON, K. AND MOON S. 1998. Semantic multigranularity locking and its performance in object-oriented database systems. *Journal of Systems Architecture, 44, 12,* 917-935.

This paper provides ISMGL, a semantic multigranularity locking method for OODBs, and a brief performance evaluation of it. The authors state their method to be beneficial as it exploits the commutativity of methods by enhancing the degree of concurrency and decreases locking overhead by using multigranularity locking rules. According to the authors, currently available semantic concurrency control, proposed by Muth et al. [1993] faces the commutativity presumption problem in the presence of shared subobjects. The authors claim to resolve this problem by using the in-place conflict detection approach in ISMGL.

In their database model, the authors have assumed that an object can participate in more than one complex object. Two methods defined on different classes are not thought to be compatible in their model. They portray this as the methods can be in conflict in the presence of shared sub objects and

the conflict relationships would be determined dynamically. In this model, compatibility means that they could execute concurrently without damaging the database consistency. For each class, a commutativity table showing the compatibility relationships between methods is assumed and considered as a special property of the class.

The authors propose a concurrency control protocol that they claim to synchronize the execution of nested transactions with exploiting method commutativity and a multigranularity locking scheme. They have used the in-place conflict detection to solve the commutativity presumption problem. It detects a conflict between methods which might access shared subobjects without explicitly specifying commutativity relationships among all the methods defined on all classes. This process in done only when component objects of a complex object is locked individually. This method decides whether a method-level conflict occurs only when the actual low-level conflict took place.

The authors claim the correctness of ISMGL as it allows only semantically serializable execution. Because it uses multigranularity locking and nested transaction along with commutativity of methods, the authors declare their approach to be a novel one. Besides, from the results of performance evaluation they indicate that the integration of a multigranularity locking and a semantic-based concurrency can give better performance in OODBs.

- This work is related to:

MUTH, P., RAKOW, T. C., WEIKUM, G., BROSSLER, P.AND HASSE, C. 1993. Semantic Concurrency Control in Object-Oriented Database Systems. In *Proceedings on IEEE 9th International Conference on Data Engineering,* 23-32.

RESENDE, R. F., AGRAWAL, D. AND ABBADI, A. E. 1994. Semantic Locking in Object-Oriented Database Systems. In *Proceedings of OOPSLA 94,* 388-402.

- This work was later referred by:

JUN, W. 2000. Semantic-based locking technique in object-oriented databases. *Information and Software Technology, 42,* 8, 523-531.

JUN, W. 1998. An integrated concurrency control in object-oriented database systems. *PhD Thesis, University of Oklahoma.*

JUN, W. AND GRUENWALD, L. 1998. Semantic-Based Concurrency Control in Object-Oriented Databases. *Journal of Object-Oriented Programming, 10,* 8, 33-39.

This paper intends to address three important issues in OODB, namely semantics of methods, nested method invocation and referentially shared object. According to the authors, the existing methods cannot handle them properly and these issues need to be taken care of as in OODB; the semantics of methods on encapsulated objects can be exploited to provide better concurrency and a method invoked on a higher level object can invoke another method defined on a component object of the higher level object. Again, referentially shared objects are very common in OODB and for them conflict relations should be determined for dynamic method execution.

For their concurrency control method, the authors have assumed the objects to be organized in a hierarchy and referential sharing has been allowed. Also, a method in an object is assumed to be able to invoke methods on objects which are lower in the hierarchy. Their method is based on two-phase locking. When a transaction invokes a method on an object, a lock is required for it. They have also presented a way to automate commutativity relationships among methods which is based on the notion of affected set of attributes. So, even if two methods conflict, as long as their access modes on individual attributes do not conflict, those two methods can run in parallel.

For dealing with the three aspects of their proposed method, first, they allowed the application programmers to define commutativity relationships for some methods by using semantics of methods. So, two methods can commute semantically at the discretion of an application programmer even though they do not commute in terms of read and write access modes. For nested method invocations, they have associated each method invocation with a lock. Before any method invocation, a lock is requested and granted and when it is finished, the lock is inherited by its parent. If a transaction is finished, its locks are discarded. Again, for referentially shared objects, method invocations on different objects may lead to conflicts. So, the authors have determined the conflicts dynamically for each sub object.

The authors claim their protocol to be correct, as for every execution resulting from their protocol; they could find an equivalent serial execution by a series of two transformations: substitution and commutativity-based reversal. They admit not to consider inheritance hierarchy and refer an integrated protocol for both object and inheritance hierarchy as their future work.

- This work is related to:

AGRAWAL, D. AND ABBADI, A. E. 1992. A Non-restrictive Concurrency Control for Object-Oriented Databases. In *proceedings of 3rd International Conference on Extending Database Technology*, 469-482.

CART, M. AND FERRIE, J. 1990. Integrating concurrency control into an object-oriented database system. In *proceedings of 2nd International Conference on Extending Data Base Technology*, 363-377.

RESENDE, R. F., AGRAWAL, D. AND ABBADI, A. E. 1994. Semantic Locking in Object-Oriented Database Systems. In *Proceedings of OOPSLA 94*, 388-402.

- This work was later referred by:

JUN, W. 1998. An integrated concurrency control in object-oriented database systems. *PhD Thesis, University of Oklahoma.*

GRESTL, D. S. 1998. Semantic Concurrency Control Recovery and Performance Profiling for Improving Response Time in Database systems. PhD thesis, State University of New York.

This dissertation intends to demonstrate that semantic correctness criteria can be applied to the design of a concurrency control algorithm and proposes a scheme where transactions are decomposed depending on semantic criteria and then scheduled using the Assertional Concurrency Control (ACC). The authors claim that in an environment where lock contention hampers the system, ACC can provide a better performance and their use semantic correctness instead of serializability provides more concurrency compared to a two-phase locking. They have taken this approach as serializability tremendously decreases the flexibility of solutions to the performance problems of two-phase locking.

According to the authors, ACC can work when the programmer and the concurrency control have full knowledge of all the transactions of the system. The ACC is an extension of the conventional concurrency control system and the transactions of the system are divided into atomic, isolated steps and the scheduling of these steps is done as per the user provided specifications. If this decomposition is not free of residual interference, then ACC manipulates step interleaving by efficiently locking the interstep assertions. This in turn ensures that the precondition of a step remains true while initiation of it occurs.

[Bernstein et al. 1998] introduced one-level ACC, which integrates ACC within the database concurrency control. This dissertation is an effort to modify this ACC algorithm to further improve performance. The current ACC system allows multi-step transactions to be executed on a single-server and

uses a modified ahead log algorithm. Step commits are logged and this record identifies the corresponding compensating step for it. If compensation is found to be necessary, the current step is aborted and compensation is invoked. It can handle crash recovery by aborting all in-progress steps and running compensation for the on-going transactions.

After running some experiments, the authors claim that ACC outperforms the two- phase locking system if the amount of contention is high and the benefit is in a correlation to the amount of contention. The authors admit that ACC causes some extra overheads like, extra time for engine to process lock conflicts and for execution of pre-step commit logging and the execution of recovery. As a future work, they mention further improvement of the measurement algorithm and the analysis algorithm of ACC.

- This work is related to:

BADRINATH, B. AND RAMAMRITHAM, K.1992. Semantics-based concurrency control: beyond commutativity. *ACM transactions on database systems, 17,* 163-199.

BERNSTEIN, A. AND LEWIS, P. 1996. High performance transaction systems using transaction semantics. *Distributed and parallel databases, 4,* 1, 25-47.

WEIKUM, G. 1991. Principles and realization strategies of multilevel transaction management. *ACM transactions on Database Systems, 16,* 1, 132-180.

- This work was later referred by:

No relevant papers were found.

MUNNICH, A. 2000. PRED-DF - a data flow based semantic concurrency control protocol for real-time main-memory database systems. In *proceedings of the Seventh International Conference on Real-Time Systems and Applications (RTCSA'00),* 468-472.

This paper describes a new semantic concurrency control protocol called PREDeclaration and Data Flow Analysis based Concurrency Control protocol (PRED-DF) for main-memory real-time database systems. According to the author, most of the available real-time concurrency control protocol methods aim to optimize disk-based systems and gives attention towards performance improvement. Moreover, these methods are not suitable for hard real-time systems as they are not predictable and interfere with task scheduling. To resolve these issues, the author proposes this new mechanism.

PRED-DF is a locking based model which generates uses serializable schedules. The basis of this protocol is a pre declaration protocol. It also uses the information gained from the analysis of the application's data flow in order to improve concurrency. The data flow analysis is done with Transaction Conflict Graph (TCG). Any cycle within the TCG is a problem for concurrency control and schedules containing such cycles are prevented by PRED-DF. The protocol, PRED-DF has two consecutive steps. In the first step, all information that is obtained in advance is extracted. In the next step, the online concurrency control decisions are made, transactions are scheduled online according to a locking protocol. Here, in this method, unlike traditional concurrency control mechanisms, locks are released as soon as possible and so PRED-DF can handle many situations which the conventional ones are incapable of. Moreover, the author claims PRED-DF to be deadlock free and able to produce serializable schedules.

The proposed protocol is said to be implemented as part of the real-time database prototype, Component Based Active Real-time Database. The database is implemented as a library initially and later linked with an application. The locking mechanism associated with it is based on conflict counters and conflict lists. Thus, the authors state to save the CPU time for locking the single data objects or groups of data objects.

The author claims to provide predictability or the advance deadline verification, low cost due to the high efficiency of predeclaration with large lock granularities, less pessimistic deadline verification because if short locking times and low cost as the locking is done on whole data sets and there is no abort or rollback of transactions in this method.

- This work is related to:

D. AGRAWAL, J. L. BRUNO, A. E. ABBADI, AND V. KRISHNASWAMY. 1994. Relative Serializability: An Approach for Relaxing the Atomicity of Transactions. In *Proceedings of the 13th Principles of Database Systems,* 139-149.

C. PU AND A. LEFF. 1992. Autonomous Transaction Execution with Epsilon Serializability. In *Proceedings of 2nd International Workshop on Research Issues on Data Engineering: Transactions and Query Processing,* 2-11.

- This work was later referred by:

No relevant papers were found.

MADRIAA, S. K., TUBAISHATB, M. A., BHARGAVAA, B. 2000. Multi-level transaction model for semantic concurrency control in linear hash structures. Information and Software Technology Journal, Elsevier Science, 42, 7, 445-464.

This paper exploits the semantics of the linear hash operations at each level of transaction nesting using multi-level transaction model to increase concurrency. The authors claim to design the system using layered system architecture in a three-tier client/server environment to allow more flexibility in decomposing of application programs. For implementing their idea, they have used a model using object-oriented technology and multithreading concept.

In this paper, the authors represent each linear hash operation by a sequence of operations at lower level of abstraction and each of the linear hash operation at leaf-level is a combination of search and read/write operations. They consider locking at two levels: bucket and key level, which in turn provides more concurrency. In their model, for aborts of restructuring operations, no compensated transaction is used as they found that these kinds of operations can affect the structure of the data, but not the key values stored in them.

The authors claim to treat buckets (pages) as objects and linear hash operations as methods. Each object is given an object id to distinguish it from other objects. With each method invocation, a message is created and sent to other objects to the same object. The control code enables an object to access its data with the help of one of its methods or operations. A sender passes a message to a receiving object and asks the object to perform a task. The receiving object may return the information to the sender or pass messages to other objects requesting the execution of additional tasks. The methods used in their model is supposed to correspond to multi-level transactions and are implemented as multithreads as it can implement the behaviors of multi-level transactions in an efficient way. Their use of layered system architecture in three-tier client/server environment so that it can allow more flexibility in terms of decomposing of application programs whose modules may be designed and implemented independently.

The performance evaluation of the algorithm to examine the probable increase in concurrency and the overheads associated with it is mentioned as a future work. The authors plan to study the behavior of their model in a distributed environment, too.

- This work is related to:

ELLIS, C. S. 1987. Concurrency in linear hashing. *ACM Transactions on Database Systems 12, 2,195–217.*

MOHAN, C. 1993. Aries/LHS: a concurrency control and recovery method using write-ahead logging for hashing with separators. In *Proceedings of the Ninth IEEE International Conference on Data Engineering.*

WEIKUM, G.1991. Principles and realization strategies of multi-level transaction management. *ACM Transactions on Database Systems 16, 1, 132–180.*

- This work was later referred by:

No relevant papers were found.

KANGSABANIK, P., MALL, R., MAJUMDAR, A. K. 2000. Semantic Based Concurrency Control of Open Nested Transactions in Active Object Oriented Database Management Systems. *Distributed and Parallel Databases, 8,* 2,181-222.

This paper introduces a concurrency control scheme for open-nested transactions in Active object oriented databases (AOODBMS) and is suitable for co-operative and long duration activity management. According to the authors, they have utilized the semantics of the transactions and thus achieved better cooperation and concurrency among the transactions.

In case of AOODBMS, it is impossible to determine whether an execution of a set of transactions is “correct” or not in presence of detached mode ECA rules and inter-transactional events. The authors have addressed this problem. They have treated atomic transactions of AOODBs as base transactions and a collection of base or complex transactions, a set of detached mode ECA rules and a state transition model. This state transition model is assumed to specify the correct execution sequences for base or complex subtransactions.

The authors of the paper have defined CoopComp-schedule, a correct class of schedule that satisfies the state transition semantics of the individual complex transactions within the schedule and maintains cooperation and compensation semantics of the individual complex transactions. The authors claim that their concurrency control method, called NP-QuadLock generates only CoopComp-schedules and helps in defining the execution semantics of AODBMS transactions.

For generating CoopComp-schedules of the inter-leaved execution of a set of complex transaction tree, the authors have used a two level scheduling, in which the first level of scheduling is done by the individual complex

transactions whereas the second level maintains the cooperation semantics within the generated schedule. The locking scheme of NP-QuadLock uses four modes of locks: shared, exclusive, relatively shared and relatively exclusive. Among them, shared and exclusive locks are similar to those used in two-phase locking. On the other hand, relatively shared and relatively exclusive locks produce non-serializable, cooperative interleaving among complex transactions. The scheme designed by them is similar to that of Farrag and Ozsu [1989].

The proof of correctness of the proposed scheme is given and the scheme has been implemented too.

- This work is related to:

BERNSTEIN, P., HADZILACOS, V. AND GOODMAN, N. 1987. Concurrency Control and Recovery in Database Systems, *Addison Wesley.*

BUCHMANN, A., ZIMMERMAN, J. BLAKEY, J. AND WELLS, D. 1995. Building and integrated active object oriented database management system: Requirements, architecture and design decisions. In *Proceedings of 11th International. Conference on Data Engineering.*

- This work was later referred by:

No relevant papers were found.

JUN, W. 2000. Semantic-based locking technique in object-oriented databases. *Information and Software Technology, 42*, 8, 523-531.

This paper talks about a locking-based semantic concurrency control that deals with semantics of methods, nested method invocation and referentially shared object of object-oriented database. Apart from this, a performance study has also been conducted by simulation and then compared with two existing techniques.

[Jun and Gruenwald 1998] proposed a concurrency control scheme addressing the same issues. In that method, they have used locks for the execution of methods rather than atomic operations, which can reduce the locking overhead and deadlocks. They provided a way of automating commutativity of methods and increased the concurrency by using run-time information. This paper extends their work by doing a simulation to get a performance comparison study with the existing methods: Orion and Malta's. The paper also claims the proposed scheme to be better based on the results of that study.

The simulation model used by the author is based on the model used in the existing concurrency control performance evaluation and is implemented using SLAM II simulation language. The author justify taking up these two methods for evaluation as both Orion and Malta do not consider parent child parallelism; locks are required for every atomic operation in Orion and for each method invocation in Malta. The simulation model comprised of six components: Transaction generators, transaction manager, CPU scheduler, lock manager, deadlock manager and buffer manager and two physical resources: CPU and memory. The author used two test cases for nested method invocations: by varying the arrival time and the nested method invocation to the non-nested method ratio and examined the changes with respect to transaction response time for all three methods.

From the analysis of the results, the author shows that while varying arrival rate for nested method invocations the transaction response time of the proposed technique is 30.8% lesser than Orion and 16% lesser than Malta's. Again, the testing case of varying nested method invocation to non-nested method invocation ratio shows that the proposed scheme has 28.3% lesser transaction time than Orion and 11.2 % lesser than that of Malta's.

The author refers the extension of the proposed scheme for evolving OODBs and for distributed OODBs as the future work.

- This work is related to:

KWON, K. AND MOON, S. 1997. Semantic multigranularity locking for object-oriented database management systems. *Journal of Database Management Systems, 8,* 2, 23-33.

RESENDE, A. F., AGRAWAL, D. AND ABBADI. A. E. Semantic locking in object-oriented database systems. In *Proceedings of OOPSLA'94,* 388-402.

- This work was later referred by:

No relevant papers were found.

NETO, P.F.R., PERKUSICH, A., PERKUSICH, M.L.B. AND TURNELL, M.F.Q.V. 2001. Analysis of periodic transactions and semantic concurrency control for real-time databases using colored Petri nets. In *proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 4,* 2723-2728.

This paper deals with introducing a method for analyzing the semantic concurrency control technique applied in real-time database applications as well as the scheduling of periodic transactions with hard deadlines. The

modeling method used for this purpose was based on object-oriented concepts and colored Petri nets.

DiPippo [1995] proposed semantic lock technique, a semantic concurrency control method that allows data and transactions' logical and temporal consistency negotiation. In this method, the concurrency control is distributed amongst the objects and a compatibility function. A compatibility function is defined for a pair of methods than can execute concurrently for any given object. The authors of this paper tried to examine whether the compatibility function was correctly defined by the designer or not and that required a Colored Petri Net to be built.

For analyzing the semantic concurrency control and scheduling of transactions a finite set of real-time transactions were modeled in Colored Petri Net. For each transaction there were four temporal parameters defined: transaction liberation, transaction computation time, maximum transaction execution period and periodicity of transaction. According to the authors, analysis process enables to prove that all transactions can commit satisfying the defined deadlines.

For the formal analysis process, the general properties of the model representing the required behavior of the system have to be defined and the specification is to be used to prove the properties. Occurrence graph, the state space representation of a colored petri net, has been used for this purpose. Once the state space is built, the graph is searched for valid scheduling sequences and at the end, the best one is chosen.

The authors claim that this analyzing method verifies whether the compatibility functions defined for the semantic concurrency control is correct or not and introduces a technique depending on the state space of the colored petri net model which can find the best scheduling for the periodic transactions with hard dead-lines for a real-time database application. The authors state that investigating other concurrency control mechanisms of real-time database system by applying the approach of this paper, is their future work.

- This work is related to:

BESTRAVOS, A. LIN, K.J. AND SON S.H. 1997. Real-time database systems: Issues and applications. *Kiuwer academic publishers.*

DIPIPPO, L.C. 1995. Semantic real-time object based concurrency control. *PhD thesis, Department of computer Science and statistics, University of Rhode Island, 96-247.*

JENSEN, K. 1998. An Introduction to the practical use of coloured petri nets. In *lectures on Petri nets II, applications. Springer.*

- This work was later referred by:

NETO, R., PERKUSICH,P. F., PERKUSICH, A., BARBOSA, M. L. 2003. Scheduling and Semantic Concurrency Control Analysis for real time databases. In *proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications.*