

---

# Customizable Isolation in Transactional Workflow

Adnene Guabtni, François Charoy, and Claude Godart

LORIA laboratory, BP 239, 54506 Vandœuvre-lès-Nancy Cedex, France  
{guabtni,charoy,godart}@loria.fr

**Summary.** In Workflow Management Systems (WFMSs) safety of execution is a main need of more and more business processes and transactional workflows are real needs inside enterprises. In previous works, transactional models consider mainly atomicity as the main issue regarding long term transactions. It rarely consider the fact that many processes may run concurrently and thus access and update the same data. Usually, the main isolation item is the data on which we apply locking approaches and this attitude don't worry about process dimension. In this work we study more precisely what are the real isolation needs in workflow environment. To realize these needs, we define "Isolation Spheres" inspired from "spheres of control" proposed by C. T. Davies to make a separation of concerns between workflow design and transactional properties specification.

## 1 Introduction

Defining the transactional requirements of business processes is still an issue in today workflow models and systems. This is even more critical when the complexity of the process increases. It is the case for instance with cooperative process or with distributed and composed e-services. Today's models consider the relationship between transactional properties and processes as something very monolithic. A process is considered as a long term atomic transaction and an activity is considered as a short term transaction. In the workflow terminology, that means that a process is controlled by some kind of advanced transaction model that ensure either that the process terminates or that it can be compensated. The other assumption is that activities can be implemented as short term database transactions. This has an impact on the way processes and activities are defined and it requires that business process designer have some in-depth knowledge of transactional requirements. Moreover these models consider mainly atomicity as the main issue regarding long term transactions. It rarely considers the fact that many processes may run concurrently and thus access and update the same data. Some work has been done

on this topic in a recent past (contracts/coo) but it has never been generalized to process.

In this paper we try to consider processes as the concurrent execution of sets of activities that may have different requirements regarding isolation. Usually, isolation in workflow systems is performed by the database system of WFMS. Databases use ANSI SQL isolation levels to define isolation requirements of a transaction on some data items. The problem is that workflow isolation requirements cannot always be satisfied by a database system. Contrary to database transactions, workflow transactions are defined and organized through a process. At design time, we know exactly what are possible concurrent transactions and we want to make it possible to allow a transaction to adopt different isolation levels depending on concurrent transactions. This need appears when an activity requires an isolation level to access some data and many activities become unable to access or modify this data even if some of them don't really affect the transaction's correctness or the consistency of data. The way we choose to tackle this problem consists in separating concern between the process definition and its transactional requirements. We consider that a process must be defined independently of the transactional properties that we need to ensure. The process definition depends on the actual user activities and should reflect the actual company organization. Transactions reflect technical and consistency requirements and should not impact on this definition. To perform that, we inspire ourselves from the sphere of control approach proposed by C. T. Davies in [5] in 1978. This approach has been reused in 2001 to produce atomicity spheres in [4] to perform customizable atomicity specification in transactional workflow. We use the same approach to define isolation spheres to allow customizable isolation specification.

In the following sections of the paper, we study isolation needs in database world already applied to workflow processes. Next we try to specify transactional workflow isolation requirements. Finally we develop our approach based on isolation spheres to allow customizable isolation in transactional workflow.

## 2 Isolation needs in transactional workflow

Isolation is an important and difficult problem as it requires to consider access to data during process and activities execution. It requires to study data manipulation by process, activities and/or sets of activities. It requires also to take into account the fact that long term process execution cannot require locking of whole set of data for all its duration. The requirements regarding these data can be far more complex. Isolation levels in flat transactions has been recognized in ANSI SQL specification [1] where the user can choose between 4 different isolation levels : (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE) to prevent phenomena like dirty read, fuzzy read or phantom problems as described in the table below. Dirty read problem occurs when a transaction reads an uncommitted

data rolled back later. Non repeatable or fuzzy read problem occurs when a transaction reads a data two time and retrieve two different values. Phantom problem occurs when a transaction reads a set of data satisfying some search condition and then repeats its read with the same search condition, it gets a different set of data items.

Isolation levels	Dirty read	Fuzzy read	phantom
READ UNCOMMITTED	Possible	Possible	Possible
READ COMMITTED	Not possible	Possible	Possible
REPEATABLE READ	Not possible	Not possible	Possible
SERIALIZABLE	Not possible	Not possible	Not possible

**Table 1.** SQL isolation levels defined in terms of the three phenomena

WFMS and Databases don't refer to the same requirements. Workflow processes are based on a controlled flow of tasks but this control is not sufficient to ensure correct execution and don't prevent from lack of consistency. This is due to workflow data visibility that is a paramount way to distribute access to data in a workflow process but also a real source of concurrency access. In the next section, we expose isolation needs in workflow processes and what is important to do in the case of activities groups.

## 2.1 Isolation needs in WFMS

Data accessed during a process execution are heterogeneous. They consist in documents, folders, cases data, local data, database system data and/or data obtained from external sources. Access control on these data may be very different and may have different kind of impact on the level of isolation that can be obtained. Moreover access to these data can be controlled by automatic activities or by users themselves. The level of control differs also in these two cases. Execution of automatic programs can be anticipated. User action cannot. We need to take all these parameters in account to study isolation requirements in workflow processes.

Based on previous conclusions, we need to introduce new elements in the isolation levels used performed by the transactional workflow designer. These elements are the cohesion and the coherence on a group of activities. In the following, we describe these new workflow isolation behaviors

One of the needs of transactional workflow is the control of the **cohesion** of data used in a group of activities (collaborative work, distributed or composed E-services, ). The solution used nowadays to ensure this cohesion of data for groups of activities is to create only one transaction imbricating all the others.

Admittedly this approach makes it possible to ensure such a cohesion but has a major impact on the competition of access since it calls upon bolts in writing.

A second need is that of the **coherence** of the data. Indeed, the fact of allowing activities external to a group to read some data written by activities of that group. This can cause some inconsistencies outside the group. that's why a control of the data visibility written by activities of a group must be ensured.

Related works were made in [6] to support partial isolation in flat transactions but it was made without a real separation of concerns. In the reality, relativity and extension of isolation are merged to express customizable requirements. These requirements are usually influenced by the requirements of each activity and the pertinence of the isolation is more and more crucial depending on the type of used data and its visibility in the workflow [7]. In the next section, we introduce a new approach based on "isolation spheres" to take into account workflow isolation needs expressed in this paper.

### 3 Our approach : Isolation Spheres

In the last few years, some works has been inspired from the sphere of control proposed by Davies [5] to enhance expressivity of transactional properties, especially in [4] where the notion of atomicity sphere has been developed to allow more customizable atomicity in transactional workflow. A sphere of atomicity is a group of activities on which we apply the transactional property of atomicity. In our work, we inspire from this sphere of control approach and we define "spheres of isolation". A sphere of isolation will allow us to generalize isolation in the context of a workflow system. An isolation sphere allows the inside group of activities to be isolated from concurrent outside activities. The level of isolation is defined by the sphere. two kind of constraints are defined by the sphere : Coherence and Cohesion.

An isolation sphere controls the access to some data giving some privileges to a set of activities and some others to the rest of workflow activities depending on the execution evolution inside the set.

An isolation sphere represents a set of activities in concurrency working on some data. All or a part of this data represents the isolation data (data concerned by isolation on which necessary locks need to be applied). To perform cohesion and coherence of this data, we introduce some cohesion levels and some coherence levels :

**Read Uncommitted** : if an activity of the sphere reads a data, it can read only the last value written before the starting of the sphere or a value written by an activity of the sphere. Thus, the group of activities constituting the sphere starts from the same value.

**Read Committed** : if an activity of the sphere reads a data then it can read only the last **validated** value written before the starting of the sphere or a value written by an activity of the sphere.

**Repeatable Read :** As the Read Committed except that it is also concluded that the value of the data is not modified by an external activity as long as the sphere did not finish its execution yet. The end of the execution of a sphere occurs when all its activities finished their execution.

**Serialisable:** emulate an execution in series of the activities of the sphere with outside ones. This level makes it possible to ensure a serialisability between the sphere and the rest of the process but does not ensure it between the activities of the sphere.

To ensure Coherence, some coherence levels are defined in the following :

**Atomic coherence :** All the values of a data written by the activities of the sphere are visible outside of the sphere.

**Selective coherence :** Only the **validated** values written by the activities of the sphere are visible outside of the sphere.

**Total coherence :** Only the **last validated** value written by an activity of the sphere is visible outside of the sphere.

Imbrication of isolation spheres is possible throw imbrication of sets of activities. Imbrication is a powerful way to express more possibilities in isolation behavior. While isolation levels can be relative to a part of the process, we can generate isolation behavior dependent on execution progress due to isolation relativity over sphere imbrication.

The power of isolation spheres is the simplicity of interpretation : an isolation sphere is represented as a group of activities that need to be isolated from external activities and don't worry about internal concurrency (concurrency between activities of the group). Internal isolation, if needed, can be performed by imbricated isolation spheres. So the work performed by the workflow designer to specify isolation requirements is simplified.

This isolation sphere based transactional workflow take account of more possibilities to customize isolation and introduce more flexibility in isolation behavior. But isolation levels defined in the ANSI SQL specification have been criticized in [3] due to the lack of clarity in the interpretation of these isolation levels and the lack of response to some phenomena other then dirty read, fuzzy read and phantom. Since that, ANSI SQL specification has changed to be SQL 3 but without changes in isolation levels. Non SQL isolation levels have been proposed as cursor stability isolation or snapshot isolation. We need to study the impact of using these isolation approaches on the isolation sphere definition.

## 4 Conclusion and perspectives

In this paper, we have focussed on isolation in transactional workflow. Existing approaches use techniques of isolation adapted to databases and this practise is not really adapted to workflow context. A specific adaptation of isolation techniques to transactional workflow increases expressivity in term of isolation and allow process to get rid of long blocking due to database isolation methods.

Our study of the problem revealed two main isolation functionalities to make part of the transactional workflow possibilities : Cohesion to make possible the activities of a group to start working from the same values of data and become unified along the sphere execution, and Coherence to make it possible to control the delivery of data values to external activities. Our approach to make these two functionalities possible is based on "Isolation Spheres" inspired from Sphere of control introduced by C. T. Davies.

This work need to be continued referring to many aspects : the relation between isolation spheres declaration and the control flow of the workflow process, the correctness of imbricating spheres and the flexibility criterion that we need to find to ensure that a transaction will be performed with less blocking then before. Also an implementation of "isolation sphere" functionalities need to be performed in a WFMS to validate the feasibility of this work.

## References

1. Ansi x3.135-1992, american national standard for information systems - database language - sql. November 1992.
2. W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51, 2003.
3. Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O’Neil, and Patrick O’Neil. A critique of ansi sql isolation levels. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 1–10. ACM Press, 1995.
4. Wijnand Derks, Juliane Dehnert, Paul Grefen, and Willem Jonker. Customized atomicity specification for transactional workflow. In *Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications (CODAS’01)*, pages 140–147. IEEE Computer Society, 2001.
5. Charles T. Davies Jr. Data processing spheres of control. *IBM Systems Journal* 17(2): 179-198, 1978.
6. Randi Karlsen. Supporting partial isolation in flat transaction. In *Proceedings of the 10th International Conference on Database and Expert Systems Applications*, pages 698–711. Springer-Verlag, 1999.
7. Nick Russell, Arthur H. M. ter Hofstede, David Edmond, and W.M.P. van der Aalst. Workflow data patterns. Technical Report FIT-TR-2004-01, Queensland University of Technology, Brisbane, Australia, April 2004.