

## Spheres of isolation: adaptation of isolation levels to transactional workflow

Adnene Guabtni, François Charoy, Claude Godart

► **To cite this version:**

Adnene Guabtni, François Charoy, Claude Godart. Spheres of isolation: adaptation of isolation levels to transactional workflow. Third International Conference on Business Process Management - BPM'2005, LORIA - Laboratoire Lorrain de Recherche en Informatique et ses Applications, Sep 2005, Nancy/France, pp. 458 - 463, 10.1007/11538394 . inria-00000246

**HAL Id: inria-00000246**

**<https://hal.inria.fr/inria-00000246>**

Submitted on 19 Sep 2005

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Spheres of isolation: adaptation of isolation levels to transactional workflow

Adnene Guabtni<sup>1</sup>, François Charoy<sup>2</sup>, and Claude Godart<sup>3</sup>

<sup>1</sup> guabtni@loria.fr

<sup>2</sup> charoy@loria.fr

<sup>3</sup> godart@loria.fr

LORIA - INRIA - CNRS, BP 239, 54506 Vandœuvre-lès-Nancy Cedex, France

**Abstract.** In Workflow Management Systems (WFMSs), transaction isolation is managed most of the time by the underlying database system using ANSI SQL strategies. These strategies do not take sufficiently into account process aspects. Our work consists in studying with more depth the relation between isolation strategy and process dimension as well as the real isolation needs in workflow environments. To carry out these needs, we define ‘spheres of isolation’ inspired from ‘spheres of control’ proposed by C. T. Davies. Spheres of isolation take into account real workflow isolation needs with separation of concerns between workflow design and the specification of its transactional properties.

## 1 Introduction

The specification of transactional constraints in business processes is always a paramount stake especially in co-operative processes or distributed and composed e-services. In WFMSs, transactions are usually implemented by the DBMS. Those systems generally use standard ANSI SQL [1] to define the isolation’s constraints of a transaction. The problem lies in the fact that these isolation’s constraints cannot always satisfy those of a workflow process. The process dimension in atomicity has been already analysed in [4] and give more capabilities to transactional WFMS. Isolation [11] has been already studied in cooperative process environment in a recent past (Contracts [9] and Coo [6]) but has never been generalized to workflow processes.

To carry out that problem, we take as a starting point the approach of ‘Spheres of control’ proposed by C.T. Davies in [8]. This approach was re-used in [4] to introduce spheres of atomicity allowing customised specification of atomicity in transactional workflow. We follow the same approach to define spheres of isolation in order to allow a customized specification of isolation constraints in transactional workflow. We consider a process as being the concurrent execution of sets of activities which can have various constraints regarding isolation. We want to allow the workflow designer to decide on the degree of isolation necessary for a group of activities. Our approach introduces also a separation of concerns between process and transactional properties definition. The definition of the process should reflect the real organization of work in the company.

Transactional properties should reflect technical aspects of the execution and consistency needs and should not influence the process definition.

In the following sections of the article, we analyze the stakes and needs of isolation in workflow systems compared to database systems. Next we develop our approach based on ‘spheres of isolation’ to allow customized isolation in transactional workflow.

## **2 Transactions in workflows : current approaches**

Advanced transaction models were introduced to enhance transaction support in WFMSs and provide more flexibility compared to traditional database transactions (ACID). Their implementation in workflows was studied in [2]. These models included process dimension on transaction management but were focused mainly on atomicity property. Current implementations of transaction models in WFMS are so heterogeneous and complex that a real taxonomy of transactional workflow implementations was defined in [5]. This taxonomy is a representation of the real practice of transactional properties in WFMSs.

Our approach is to study the real needs of WFMS for isolation properties. These properties are usually confused with database transactional needs. Atomicity needs in WFMS has been already established in [4]. The crucial difference between these needs and those of database systems is the definition of atomicity constraints to groups of activities called spheres of atomicity. In the next section, we perform a similar approach to study the real isolation needs in WFMSs.

## **3 Isolation requirements in WFMS**

### **3.1 Isolation levels in traditional transactions**

The isolation problem occurs when several transactions access to the same data. In several information systems, the isolation stake grows when the data used are accessed by more and more concurrent transactions and increasingly independent transactions. The problem that occurs in this case is the lack of flexibility of the isolation strategy. In database systems, isolation is guaranteed via isolation levels [1]. There are four isolation levels: Read Uncommitted, Read Committed, Repeatable Read and Serializable. These levels make it possible to provide more or less undesired phenomena (dirty read, fuzzy read and phantom [1]).

Isolation levels suggested in (ANSI SQL, 1992)[1] were criticized in [3]. However they are largely used in current databases systems. Other approaches based on timestamps were studied and are based on optimistic locking systems. Nevertheless, all existing approaches do not express isolation requirements adapted to transactional workflows. In the next section, we expose what are process dimension based isolation requirements in WFMSs.

### 3.2 Isolation requirements for Transactional Workflow

Isolation problems are more and more obvious depending on the data visibility. Indeed, the data used in WFMS were classified in 7 types according to their visibility according to workflow data patterns[10]. The need of process support in isolation strategy depends on the workflow data visibility starting at task data visibility where there is no isolation needs and continue with this order : block, scope, case, workflow, environment, multiple instance data visibility. Multiple instance execution [7] produces the highest need of process dimension support.

The goal of our work is to adapt isolation levels to workflow. That becomes possible if we take into account not only needs of single activities but also needs of a group of activities of the process (collaborative work, distributed or composed e-services). We identified two main needs consisting in the control of cohesion and coherence of a group of activities.

Cohesion means the fact that activities of the same group use the same reference for data access. Activities can then use data with ensurance that all of them are using the same version and are seeing only changes made by them. External activities (not part of the group) that may want to modify the same data during the execution of the group will not influence the referential used by the group. The referential can be seen as a view of data, readable and writable only by activities of a restricted group.

Coherence of data is another important need. Indeed, a group of activities usually needs to ensure that the impact of its execution do not introduce some mistakes or inconsistencies. These inconsistencies are usually due to the use of temporary or uncommitted data produced by the group.

The coherence concerns the external environment of the group while cohesion concerns the internal one. Based on these two main needs, we will introduce in the next section the notion of ‘isolation spheres’.

## 4 Our approach: isolation spheres

In the last few years, some work has been inspired from the sphere of control proposed by Davies [8] to enhance expressivity of transactional properties, especially in [4] where the notion of atomicity sphere has been developed. In our work, we take the same approach to define ‘spheres of isolation’ as follows:

**Definition :** An isolation sphere represents a group of activities in a workflow process working in concurrency on some data. The sphere ensure the cohesion (constraints on reference data) and the coherence of the sphere referring to concurrent activities or other spheres. The cohesion and coherence constraints allow a process support in isolation strategy.

All or a part of the data used by sphere activities represents the data that have to be controlled (data concerned by isolation on which necessary locks need to be applied). To ensure cohesion and coherence on these data, we introduce some cohesion levels and some coherence levels. Before introducing these levels we need to define some notations:

A process (or workflow process) represents tasks called activities and these tasks are executed following an execution order established through a control flow between activities. A sphere is defined as part of a process. In WFMSs, a sphere is composed of activities of the workflow.

Let  $S$  the set of spheres and  $s \in S$ .

$\Delta_s$  is the set of data concerned by isolation sphere  $s \in S$ . This set of data is defined by the workflow designer and is a subset of the data used by the group.

$A(s)$  is the set of activities of  $s \in S$ .

The state of a data  $\delta$  changes over time due to activities execution and takes several values  $\{\delta_0, \delta_1, \dots, \delta_n\}$  corresponding respectively to several instants  $\{t_0 < t_1 < \dots < t_n\}$ . If the value  $\delta_i$  was written by an activity  $\alpha$ , we note it  $\delta_i^\alpha$ . We note  $\delta^\alpha$  the value validated (committed) of  $\delta$  written by  $\alpha$ .

#### 4.1 Properties of isolation spheres

Isolation spheres properties are cohesion and coherence. Cohesion means the fact that all activities of the sphere have the same view on data they access. The view represents a reference data that all activities of the sphere will read or update. External activities updates will not be visible from the sphere view. This common view represents the basis of cohesion in a group of activities but there are different possible cohesion levels based on the initial view isolation constraints. These levels of cohesion are as follows: Let  $s \in S$  and  $\delta \in \Delta_s$ ,

**Level 0 : Read Uncommitted :** if an activity of the sphere  $s$  reads  $\delta$  it can read only  $\max(\delta^s, \delta_i^\alpha)$  such as  $\alpha \in A(s)$  and  $\delta^s$  corresponds to the value of  $\delta$  read the first time by an activity belonging to the sphere.

**Level 1 : Read Committed :** if an activity of the sphere reads  $\delta$  then it can read only the  $\max(\delta^s, \delta_i^\alpha)$  such as  $\alpha \in A(s)$  and  $\delta^s$  corresponds to the validated (committed) value of  $\delta$  read the first time by an activity belonging to the sphere.

**Level 2 : Repeatable Read :** same case of Read Committed except that it is also concluded that the value of  $\delta$  is not modified by an activity external to the sphere 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.

**Level 3 : Serializable :** emulates an execution in series of the sphere and its external environment (activities, spheres or processes). This level ensures serialisability between the sphere and the external environment of the sphere but does not ensure a serialisability between the activities of the sphere.

Coherence of sphere represents how activities share their data with their external environment. Different levels of coherence can be defined as follows:

**Level 0 : Atomic coherence :** all the values of a data written by the activities of the sphere are visible outside of the sphere. If an activity  $\alpha$  of the sphere writes  $\delta$  then all  $\delta_i^\alpha$  are visible outside the sphere.

**Level 1 : Selective coherence :** only the **validated** values written by the activities of the sphere are visible outside of the sphere. If an activity  $\alpha$  of the sphere writes  $\delta$  then only  $\delta^\alpha$  is visible outside the sphere.

**Level 2 : Global coherence :** only the **last validated** value written by an

activity of the sphere is visible outside. If activities of the sphere  $s$  write  $\delta$  then only  $\delta^\alpha$  is visible outside the sphere,  $\alpha$  being the last activity of  $s$  to write  $\delta$ .

#### 4.2 Phenomena significance in isolation spheres context

The undesired phenomena noted in database systems don't have the same significance when we use isolation spheres. Both cohesion and coherence release isolation constraints and the significance of each phenomenon differs from a classic transaction to an isolation sphere as follows:

**For a classic transaction  $\chi$ :**

Dirty Read : Read of  $\delta_i^\alpha$  and  $\alpha$  rollbacks

Fuzzy Read : Read of  $\delta^\alpha$  such as  $\delta^\alpha < \delta_i^\beta < \delta^\chi$

Phantom : Ask for a request and the result is modified during execution by insertion of new data by another transaction

**For isolation Sphere  $s$ :**

Dirty Read : Read of  $\delta_i^\alpha$  such as  $\alpha \in A(s)$  and  $\alpha$  rollbacks

Fuzzy Read : Read of  $\delta^\alpha$  such as  $\delta^\alpha < \delta_i^\beta < \delta^s$  and  $\beta \notin A(s)$

Phantom : Ask for a request and the result is modified during execution by insertion of new data by activity external to the sphere

The control of the two dimensions (cohesion + coherence) makes it possible to define in a finer way isolation requirements for groups of activities. The choice of cohesion and coherence levels influences the degree of divergence and the degree of data exchange flexibility between activities of the sphere and its environment. Divergence increases from (cohesion3/coherence0) to (cohesion0/coherence2). Flexibility increases from (cohesion3/coherence2) to (cohesion0/coherence0).

#### 4.3 Advanced organization of isolation spheres : nested isolation spheres

Activities of a sphere are able to execute without worrying if somebody of the outside environment will obstruct their work. However it is inevitable to have requirements on isolation inside the sphere itself. A sphere can then contain others sub spheres that have different isolation needs. Thus we introduced nested isolation spheres. A sub sphere ensure its own cohesion and define its coherence with the immediate top sphere. We think that this kind of organization increases considerably the expressivity in term of isolation in a transactional workflow.

### 5 Conclusion and perspective

In this article, we have focused on isolation in transactional workflow. Existing approaches use techniques of isolation adapted to databases and not really to workflow context. We have made a specific adaptation of isolation levels to transactional workflow increasing expressivity in term of isolation and allowing process to get rid of long blocking due to database isolation methods. Our study of the problem revealed that the basic isolation entity in current transactional

workflow systems is the single activity. We have established the importance of isolation properties for groups of activities. Two main isolation properties have been established for groups of activities in transactional workflow : Cohesion and Coherence. Our approach to make these two properties realizable is based on 'Isolation Spheres' inspired from 'Spheres of control'.

This work requires to be continued in order to consider several aspects as the relation between the declaration of isolation spheres and the control flow governing the workflow, a simple way to easily choose coherence and cohesion levels, and finally an implementation of 'isolation spheres' functionalities must be carried out in a WFMS in order to validate the feasibility of this approach.

## References

1. Ansi x3.135-1992, american national standard for information systems - database language - sql. November 1992.
2. Gustavo Alonso, Divyakant Agrawal, Amr El Abbadi, Mohan Kamath, Roger Günthör, and C. Mohan. Advanced transaction models in workflow contexts. In Stanley Y. W. Su, editor, *Proceedings of the Twelfth International Conference on Data Engineering, February 26 - March 1, 1996, New Orleans, Louisiana*, pages 574–581. IEEE Computer Society, 1996.
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. Paul W. P. J. Grefen. Transactional workflows or workflow transactions? *DEXA*, pages 60–69, 2002.
6. Daniela Grigori, François Charoy, and Claude Godart. Co-flow: a process technology to support cooperative processes. *International Journal of Software Engineering and Knowledge Engineering - IJSEKE Journal*, 14(1), 2004.
7. Adnene Guabtini and François Charoy. Multiple instantiation in a dynamic workflow environment. In Anne Persson and Janis Stirna, editors, *Advanced Information Systems Engineering, 16th International Conference, CAiSE 2004, Riga, Latvia*, volume 3084 of *Lectures Notes in Computer Science*, pages 175–188. Springer, Jun 2004.
8. Charles T. Davies Jr. Data processing spheres of control. *IBM Systems Journal* 17(2): 179-198, 1978.
9. Andreas Reuter and Friedemann Schwenkreis. Contracts - a low-level mechanism for building general-purpose workflow management-systems. *IEEE Data Eng. Bull.*, 18(1):4-10, 1995.
10. 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.
11. Heiko Schuldt, Gustavo Alonso, Catriel Beerli, and Hans-Jörg Schek. Atomicity and isolation for transactional processes. *ACM Trans. Database Syst.*, 27(1):63–116, 2002.