Using Isolation Spheres for Cooperative Processes Correctness

Adnene Guabtni, François Charoy, Claude Godart

Using Isolation Spheres for Cooperative Processes Correctness

Adnene Guabtni, François Charoy and Claude Godart
LORIA - INRIA - CNRS, BP 239, 54506 Vandouvre-lès-Nancy Cedex, France
{guabtni, charoy, godart}@loria.fr

Abstract

Managing cooperation in Business Processes still represents a challenge because of several problems. Concurrent access to common data, coherence of the results, organisation and cooperation correctness are some of them. Looking at isolation problems in the database world and at their solutions using SQL isolation levels, we try to adapt them to the cooperative dimension of processes. In this paper, we identify the phenomena that happen during cooperation in business processes. Then we propose a solution based on isolation spheres to ensure correctness of cooperative processes and customise the exclusion control of the different cooperation phenomena.

1 Introduction

Today cooperative models and transaction models for cooperative environments are still focused on the implementation of the interaction and not really on the specification of its correctness. A cooperative process differs from a traditional business process on different points. First it requires being very flexible. This has been pointed out many times in the literature. Second, activities may last much longer. Developing a program module or designing a car part can take a long time. Thus, they cannot be atomic or work in isolation from other activities.

However, the requirement of activities inside a cooperative process regarding atomicity and isolation may change during the execution of the process. A process for software development for instance is composed of very flexible parts where people design, code, test and share their code and of more controlled parts, for quality assurance, validation and release. To make this distinction, we need to define what is allowed and what is not during a cooperative process. Then the process designer can define separately the process itself (workflow) and the transactional properties and especially the isolation requirements in case of cooperative processes.

To allow this separation of concerns between process design and transactional behaviour definition, we focus on aspects of cooperation that cause failures (consistency, coherence, isolation, etc). In our research we consider cooperation correctness and we focus on the identification of what can be wrong during a cooperative situation. We think that the best way to ensure cooperation correctness is to identify a complete set of phenomena that can be caused during cooperative execution and then to propose solutions for process designers to specify which correctness level they need to ensure at design time or runtime. Each level allows or not some of these phenomena. This approach is similar to the approach used for flexible transactions in database systems. Isolation levels in SQL ANSI [1] consider isolation phenomena as 'Dirty Read', 'Fuzzy Read' and 'Phantom'. These phenomena are permitted or not depending on the isolation level chosen for the transaction from 4 levels : 'Read Uncommitted', 'Read Committed', 'Repeatable Read' and 'Serializable'.

The approach of isolation levels were based on data point of view. We propose a process point of view called Isolation Spheres [5] in order to express isolation of groups of activities in a business process. A cooperation process is usually composed of cooperative parts and non cooperative parts. We try to use isolation spheres in order to ensure correctness of cooperative parts of processes. This research allows process designers to express their cooperation correctness needs without worrying about their implementation.

In the next section we expose the Isolation Spheres approach. Next we proceed with the identification of cooperation phenomena. Finally we study the contribution of Isolation Spheres to cooperative processes correctness.

2 Motivation

Advanced transactional models have been defined to cover the needs for correct business process execution. In this context, a process is considered as being a transaction with a long execution time. Activities of the process are then considered as traditional transactions with a short execution time. Activities are usually considered as basic database transactions with their associated qualities (atomicity, isolation). Thus a business process is considered from a transactional point of view as a long term transaction composed of short duration activities. Transactional properties attached to the business process are the same for all the process execution. Moreover, the transactional nature of a process is often dependent on its structure and on the activities themselves. This may not be useful and put too many constraints on the process designer. In fact, if we consider the current transactional workflow models (see the transactional workflow taxonomy [3]), the process designers must take into account transactional requirements during the design of their workflow. We argue that the process should be defined for the users and not for the transactional monitor. Our motivation is that transactional behaviour should be defined separately from the workflow design and adapted to the process dimension.

Adaptation of transactional behaviour to workflow processes has been already done for atomicity [2], [9]. Isolation in workflow processes has been considered in a recent past [8] and flexibility was carried out on this matter (Contracts [7] and Coo [4]) but has never been generalised to cooperative workflow processes and cooperative parts of processes.

In this article we consider a process as being the concurrent execution of activities which can have various constraints regarding isolation. Usually, isolation in workflow systems is ensured by the database system. 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 workflow process activities. Isolation of process activities must take into account the process organisation and workflow transaction monitors don't permit that today.

We propose Isolation Spheres as a solution to that problem. At design time and even at runtime, we can specify *a priori* which are the activities sharing the same data and which ones should be protected against concurrent access. We want to allow the workflow designer to decide on the level of isolation necessary and sufficient for these isolation spheres.

In this work, we first identify cooperation correctness needs in cooperative processes. Then we describe the isolation spheres approach to manage cooperation needs. In the next section, we develop the isolation sphere approach as a general isolation strategy for workflows, and then we identify concurrent data access problems as cooperation phenomena in cooperative processes. Finally we describe how our isolation spheres approach allows handling these problems.

3 Isolation Spheres

Isolation Spheres are inspired from the spheres of control [6]. An isolation sphere is defined by a set of activities inside a process. For these activities we want to ensure some properties regarding data accessed by the activities (Cohesion property of a sphere) and data produced by the activities (Coherence property of a sphere).

Cohesion means that all activities of the sphere have the same view on the data they access. Updates done by activities outside of the sphere must not be visible by the activities of the sphere. External activities updates will not be visible from the sphere view. This common view represents the basis of cohesion in a group of activities. Cohesion is expressed throw different cohesion levels [5] that are Read Uncommitted, Read Committed, Repeatable Read and Serialisable. These levels define the way the common view of the sphere on data is managed.

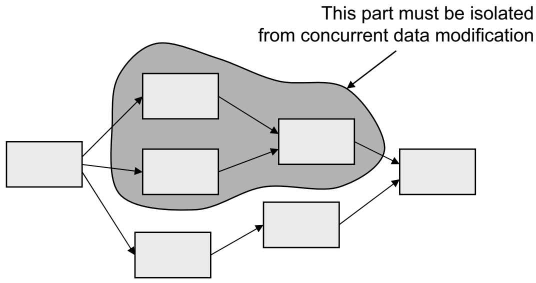

Figure 1. An isolation sphere

- **Read Uncommitted** level allows the sphere to use uncommitted values both at the start-up data view and the intermediate views during the execution of the sphere.
- **Read Committed** level allow only reading committed values also both at the start-up data view and the intermediate views etc.
- **Repeatable Read** level allows activities to read values of data with the certainty that during their use of the data, it will not change.
- **Serialisable** level emulates an execution in series without any risk of concurrency inconvenience.

Coherence of a sphere represents how activities of the sphere share their data with activities outside of the sphere. In order to control the coherence between data used by activities of the sphere and those by the rest of the processes including concurrent isolation spheres, it is essential to define a level of coherence of the sphere. Isolation spheres ensure some cohesion inside the group and also some coherence of the activities external to the sphere using the same data. The levels of coherence are the following:

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

- **Selective coherence :** Only **validated** values written by the activities of the sphere are visible outside of the sphere.
- **Global coherence :** Only the **last validated** value written by the **last activity** of the sphere is visible outside of the sphere.

4 Cooperation correctness problems in cooperative processes

Cooperation represents a crucial need in Business Processes and is usually based on data sharing between participants, tasks or services. In this research we consider safety of data use in a cooperative process context. To ease the understanding of cooperative problems we can illustrate a cooperation using the following motivating example.

4.1 Motivating example

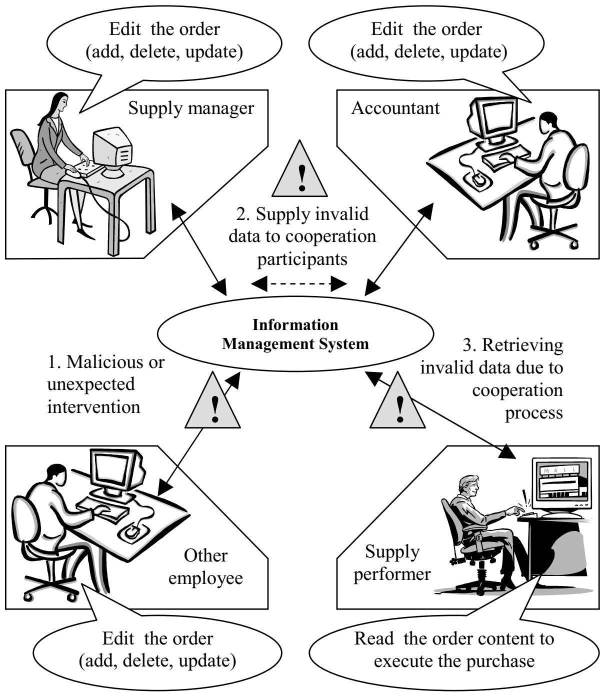

Figure 2. Motivating example

We assume a representation of a cooperative situation in a company concerning two persons working together on the elaboration of an order to purchase some products with some constraints on the number of each product, the total price of the order, the product types, the dependency relation between products etc. The order edition process consists in the work of the supply manager and the accountant. These two persons cooperate together and try to edit the order document with respect to dependencies constraints, types of products, quantities or prices. Problems that can happen while they cooperate are illustrated in figure 2 and consists of three main classes :

1. Malicious or unexpected intervention is performed by entities out of the cooperation : This is the case of an employee that introduces some modifications in the order document without being part of the cooperation between the supply manager and accountant. This intervention can induce a lack of correctness in the syntax or the semantic of the order document or simply induce a modification of the order.
2. Participant entities supply invalid data to cooperation participants : This is the case when one cooperation participant (the supply manager or the accountant in this example) delivers invalid data to the rest of the cooperation participants. For example, while the accountant is adding products to the order, the supply manager reads the current order content without worrying if the accountant has finished its edition or not. This kind of execution induce some lack of coherence in the order data.
3. Entities out of the cooperation retrieves invalid data due to cooperation : This is the case when cooperation is not clearly defined in terms of space and time. Space means the how are the participants to the cooperation and time means when the cooperation will be officially finished and so the cooperation result data is delivered. The supply performer is outside the cooperation process and don't know what exactly happen between the supply manager and the accountant. In this case of problems, He retrieves the order content before the completion of the cooperation process and uses it in the purchase process. So the supply performer will not purchase the right order.

In the next section, we study each one of the above classes in order to identify more precise phenomena during cooperative work.

4.2 Cooperation phenomena

In order to identify phenomena induced by cooperative processes, we focus on the two main properties presented by isolation spheres that are cohesion and coherence. From a cohesion point of view, phenomena are those performing perturbation of the cooperation progress and then to the cohesion of the group performing cooperation. This is the Cohesion Problematic. From this observation we can realise that the two first classes of problems detected in the motivating example take part of the cohesion phenomena.

4.2.1 Cohesion phenomena

The first class of problems untitled ‘Malicious or unexpected intervention’ provides one phenomenon that we call ‘Disrupted Cooperation’ as follows :

- **Disrupted Cooperation :** *Two activities A1 and A2 cooperate over an isolation sphere using concurrently data D. They use a value of D written during their execution by an activity not part of the sphere.*

  Activities inside the sphere reading values of the data on which they cooperate can be induced in mistake if that data is updated outside the sphere. In this case, the ” outside cooperation ” data update is not supervised.

The second class of problems untitled ‘Supply invalid data between cooperation participants’ provides three phenomena as follows :

- **Dirty Read Cooperation :** *Two activities A1 and A2 cooperate inside an isolation sphere using concurrently data D. A1 writes a value of D, A2 read it before the completion of A1 and A1 rollbacks.*

  This is similar to the Dirty Read phenomenon in Database world but in this case it is limited to the cooperation environment.

- **Fuzzy Read Cooperation :** *Two activities A1 and A2 cooperate over an isolation sphere using concurrently data D. A1 Reads a value of D, A2 write a new value of D before the completion of A1. So the work of A1 is wrong because it uses an out of date value.*

  This is similar to the Fuzzy Read phenomenon in Database world but in this case it is limited to the cooperation environment.

- **Phantom Read Cooperation :** *Two activities A1 and A2 cooperate over an isolation sphere using concurrently data of a database table. A1 requests the database with ” Where like ” conditions. A2 adds a new row to the table before the completion of A1 so that A1 uses data not up to date.*

  This is similar to the Phantom phenomenon in Database world.

Each one of the cohesion phenomena is illustrated using examples of execution schedules in figure 3

4.2.2 Coherence phenomena

From a coherence point of view, phenomena are those performing perturbation of entities out of the cooperation but caused by the cooperation progress. This is the Coherence Problematic. We can realise that the third class of problems untitled ‘Retrieving invalid data due to cooperation’

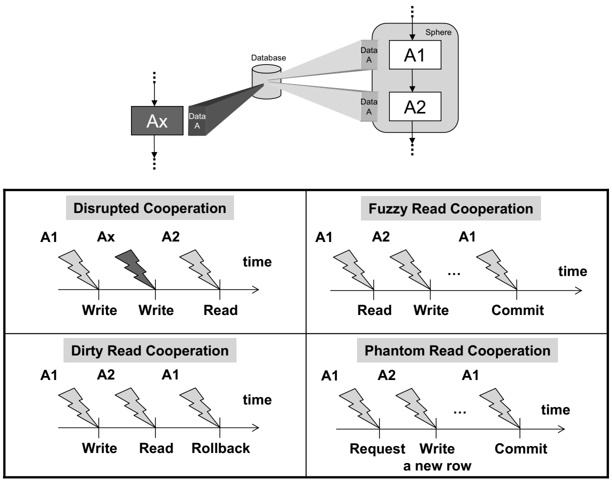

Figure 3. Cohesion phenomena and their corresponding execution schedules

and detected in the motivating example take part of the coherence phenomena and represents problems reverberated on entities out of the cooperation. The responsibility of cooperation participants in such problems consists in a lack of vigilance about delivery of invalid data or valid data but not permanent. Thus, this problem class provides two phenomena as follows :

- **External Dirty Read :** *A cooperation over a sphere inducing concurrent access to data D permits public access to uncommitted values of D written by one activity of the sphere.*

  Activities outside the sphere reading value not yet committed by an activity of the sphere can be induced in mistake if the activity that written the uncommitted value is rolled back. That’s why we call this phenomenon as ” External Dirty Read ”.

- **External Misleading Read :** *Cooperation over a sphere inducing concurrent access to data D permits public access to each committed value of D written by one of the cooperative activities.*

  Activities outside the sphere reading values committed by activities of the sphere will consider that it’s the result of the cooperation because it’s committed. That’s why we call this phenomenon as ” External Misleading Read ”.

Each one of the coherence phenomena is illustrated using examples of execution schedules in figure 4

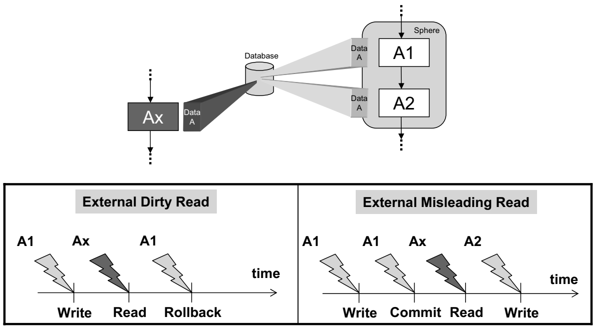

Figure 4. Coherence phenomenon and their corresponding execution schedules

5 Cooperative Process Correctness Using Isolation Spheres

Isolation spheres introduce process dimension in isolation management. Our study about this approach disclosed two dimensions : Cohesion and Coherence. The significance of each level of these dimensions is described in [5]. The duality Cohesion/Coherence of an isolation sphere, as illustrated in figure 5, expresses the Choice of each level (cohesion and coherence) depending on the needs expressed by the process designer. Depending on this choice, these levels induce more or less flexibility of the cooperative data exchanges (cohesion level) and more or less risk of divergence or incoherence (coherence level). In some cases, the process designer can accept the fact that some activities will use invalid data or not up to date data. That is the goal of these sphere level based design.

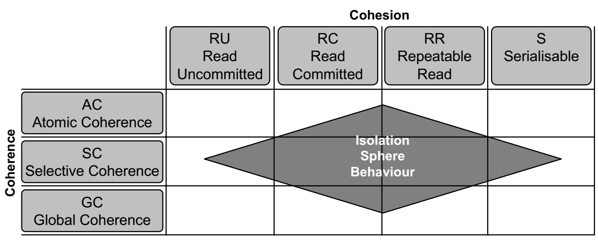

Figure 5. Duality Cohesion/Coherence : customised isolation behaviour

Applying these levels to cooperative processes allow the process designer to set the appropriate levels of cohesion and coherence for a given sphere. Figures 6 and 7 illustrate the matching between Cohesion and coherence levels with the phenomena based on cohesion and coherence problems. The process designer should take decisions about cohesion and coherence with a problem based approach build on the three main problem classes defined in the beginning of this section. This design approach allows the designer to be objective and express the best adapted levels to the situation.

	Phenomena	Disrupted Cooperation	Dirty Read Cooperation	Fuzzy Read Cooperation	Phantom Read Cooperation
No Isolation Sphere		Yes	Yes	Yes	Yes
Isolation Sphere	Cohesion level				
	Read Uncommited	No	Yes	Yes	Yes
	Read Commited	No	No	Yes	Yes
	Repeatable Read	No	No	No	Yes
	Serializable	No	No	No	No

Figure 6. Cohesion levels effects on cooperative phenomena

	Phenomena	External Dirty Read	External Misleading Read
Isolation Sphere	Coherence level		
	Atomic Coherence	Yes	Yes
	Selective Coherence	No	Yes
	Global Coherence	No	No

Figure 7. Coherence levels effects on cooperative phenomena

Cooperation phenomena →		Disrupted Cooperation	Dirty Read Cooperation	Fuzzy Read Cooperation	Phantom Read Cooperation	External Dirty Read	External Misleading Read
No Isolation Sphere		yes	yes	yes	yes	yes	yes
Cohesion level	**Coherence level**						
Read Uncommited	Atomic	no	yes	yes	yes	yes	yes
Read Commited	Atomic	no	no	yes	yes	yes	yes
Repeatable Read	Atomic	no	no	no	yes	yes	yes
Serializable	Atomic	no	no	no	no	yes	yes
Read Uncommited	Selective	no	yes	yes	yes	no	yes
Read Commited	Selective	no	no	yes	yes	no	yes
Repeatable Read	Selective	no	no	no	yes	no	yes
Serializable	Selective	no	no	no	no	no	yes
Read Uncommited	Global	no	yes	yes	yes	no	no
Read Commited	Global	no	no	yes	yes	no	no
Repeatable Read	Global	no	no	no	yes	no	no
Serializable	Global	no	no	no	no	no	no

Figure 8. Duality Cohesion/Coherence : customised cooperation behaviour

The duality Cohesion/Coherence allows us to customise the isolation strategy with twelve combinations as illustrated in figure 8. These combinations illustrate all the possibilities of flexibility in isolation in general terms and particularly cooperation. These flexibility possibilities can express the maximum degree of cooperation flexibility (Read Uncommitted cohesion level with an Atomic coherence level) but without strong data safety. or a strong data safety without any cooperation behaviour due to serialisability (Serialisable cohesion level with Global coherence level). The intermediary combinations take into account at the same time, a level to ease the cooperation work and the data exchanges between cooperation participants and another level to ensure some execution correctness degree.

The contribution of isolation spheres in terms of cooperation correctness is the choice of which cooperation phenomena to allow and which to disallow. The process designer can be sure that the cooperation process will never accept what he disallowed using isolation spheres. This kind of correctness control introduces a high flexibility level. Also, the combination concerning a sphere can be adapted during the execution to new needs and constraints following two dimensions : the levels of isolation (cohesion and coherence) and the composition of the sphere (activities that join the cooperation group and those that leave it). Modifications performed at runtime need to be coherent : if some cooperation phenomenon occurs and has been accepted by the isolation sphere levels, new levels updated by the designer at runtime must at least accept the already occurred phenomenon. The goal of this constraint is to ensure compatibility between isolation sphere levels for cooperation correctness.

6 Conclusion

In this work, we proposed to introduce process dimension in isolation strategy to ensure execution correctness in case of cooperative activities (concurrent access to common data). Our approach is based on what we call Isolation Spheres and was inspired from the Spheres of Control [6]. Cooperative processes encounter many problems in terms of transactional behaviour. In this work we identified three main classes of problems and six phenomena that can happen during a cooperation process. Based on Isolation Spheres approach, we tried to match what isolation spheres can ensure with what cooperation processes should ensure. The result is that isolation spheres deliver a complete support for cooperation phenomena exclusion with twelve levels in order to make the designer choose the balance between correctness of cooperation and flexibility in data access. Using isolation spheres for cooperative processes, the designer apply a separation of concerns between process definition and transaction requirements. Flexibility in transaction requirements definition is based on the multiple combinations of cohesion and coherence levels and can also be adapted during the execution.

This work needs more advanced studies especially in terms of correctness criteria of execution schedules and some possible algorithms to correct the execution in fault cases at runtime. A second problem is the integration of isolation spheres in BPEL (Business Process Execution Language) for web services platforms and their integration to the web services architecture. These problems are important parts of the future work and will induce the elaboration of a real integration of isolation spheres inside real workflow management systems and BPEL engines. A third problem concerns the study of levels changes at execution time and sphere composition changes and their impact on cooperation execution.

References

[1] Ansi x3.135-1992, american national standard for information systems - database language - sql. November 1992.

[2] W. Derks, J. Dehnert, P. Grefen, and W. 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.

[3] P. W. P. J. Grefen. Transactional workflows or workflow transactions? *DEXA*, pages 60–69, 2002.

[4] D. Grigori, F. Charoy, and C. Godart. Coo-flow: a process technology to support cooperative processes. *International Journal of Software Engineering and Knowledge Engineering - IJSEKE Journal*, 14(1), 2004.

[5] A. Guabtni, F. Charoy, and C. Godart. Spheres of isolation: adaptation of isolation levels to transactional workflow. In W. M. van der Aalst et al., editor, *Business Process Management, 3rd International Conference, BPM 2005, Nancy, France*, volume 3649 of *Lectures Notes in Computer Science*, pages 458–463. Springer, September 2005.

[6] C. T. D. Jr. Data processing spheres of control. *IBM Systems Journal 17(2): 179-198*, 1978.

[7] A. Reuter and F. Schwenkreis. Contracts - a low-level mechanism for building general-purpose workflow management-systems. *IEEE Data Eng. Bull.*, 18(1):4–10, 1995.

[8] H. Schuldt, G. Alonso, C. Beeri, and H.-J. Schek. Atomicity and isolation for transactional processes. *ACM Trans. Database Syst.*, 27(1):63–116, 2002.

[9] W.-J. van den Heuvel and S. Artyshchev. Developing a three-dimensional transaction model for supporting atomicity spheres. *International Workshop on Web Services Research, Standardization, and Deployment*, 2002.