Query Processing in Self-Profiling Composable Peer-to-Peer Mediator Databases

Timour Katchaounov

Uppsala University, Sweden,
timour.katchaounov@dis.uu.se

Abstract. Integration of multiple heterogeneous sources is crucial for efficient sharing and reuse of distributed data. An architecture for scalable data integration of many autonomous data sources is presented. In the architecture peer-to-peer (P2P) mediators can be defined in terms of each other through object-oriented (OO) views. Query processing with scalable performance is important to make such an architecture useful in practice. The focus of the described doctoral thesis is on query processing techniques in a composable P2P mediator architecture. Through distributed selective view expansion mediator peers are treated as 'grey-boxes' with varying level of transparency. This allows to balance between compilation time and query execution plan (QEP) quality for good overall performance. Self-profiling integrated with the query processor allows for the implementation of adaptive query processing techniques. Adaptive rebalancing of distributed QEPs based on the self-profiling capability of the optimizer detects and re-optimizes sub-optimal QEPs. The proposed P2P mediator architecture and some of the query processing techniques are implemented in the AMOS II mediator system.

1 Introduction

It is widely recognized that integration of diverse and distributed data and services is crucial to enable information sharing and reuse between people and organizations. The need for data integration occurs in many diverse contexts that depend on the degree of distribution and autonomy, the level of diversity of the information sources in terms of their data model and capabilities, the complexity of the modeled domain, the amount and dynamics of data, performance and data timeliness requirements, and the type of questions posed. Various data integration solutions are suitable depending on the combination of values for each of these parameters. One of these solutions is the mediator/wrapper approach to data integration proposed in [1].

This project is interested in the integration of a large number (> 100) of highly distributed heterogeneous and autonomous data sources with complex data. Typical environments where this type of data integration problems occur are companies producing complex engineering products, scientific communities or electronic markets to name a few. They can be characterized by many independent and distributed units ready to share some of the data and services they own, so that when combined with other sources, new valuable information is produced. This information can be used by others either to satisfy their needs or to further integrate more data, services and information to provide a higher-level integration service.

The goal of this project is to develop query compilation and execution techniques that will allow for many autonomous and distributed sources to be integrated and queried effectively through a mediator system.

Such a system should fulfill several requirements. *i)* The autonomous and distributed nature of the participating entities (e.g companies or research units) should be preserved because no one owns all data sources. *ii)* The process of data integration requires a lot of domain knowledge and is a complex and time consuming activity. It is important that this process can be scaled to large number of autonomous sources. *iii)* Each of the participants may evolve at various rate, which requires that separate parts of the system evolve independently. *iv)* A rich data model is needed to integrate complex and diverse data. *v)* Finally and most importantly a data integration system should provide high overall scalable performance.

While requirements *i) - iv)* are related more to the high-level functionality and architecture (visible to its users) of a data integration system, the last requirement *v)* is related to the internal implementation of such a system.

Therefore the research question of this project can be restated as: given an architecture that fulfills the high-level functional requirements, what are the implementation techniques that will achieve high overall scalable performance in that architecture?

To fulfill requirements *i) - iv)* a distributed peer-to-peer (P2P) mediator architecture is proposed in Sect. 3. As in [1], here mediators are relatively simple software modules that encode domain-specific knowledge about data and share abstractions of that data with higher layers of mediators or applications. Each mediator is an object-relational database system with its own storage manager, query processor and multidatabase object-oriented (OO) query language. Larger configurations of mediators are defined through these primitive mediators by composing new mediators in terms of other mediators and data sources. For the concept of a modular, sharable and distributed mediators architecture we use the term *composable mediators*. Logical composability is realized through multidatabase OO views.

Composable mediators provide the framework for the design and implementation of query processing techniques that will provide scalable performance and make the architecture useful in practice. Providing scalable performance in composable mediators poses several challenges. New techniques are needed to provide efficient query compilation and execution in a network of many logically composed mediators. Query processing should scale up to hundreds of mediator peers. In most cases it is impossible to perform precise cost and selectivity estimates when integrating many diverse data sources over a global network. This may lead to sub-optimal query execution plans. Therefore a P2P mediator system should be able to adapt to an unpredictable environment. Solutions to these issues described in Sects. 4, 5 constitute most of the contribution of this project.

There are many other research issues that are important for a successful implementation of composable mediators, such as security, automating the integration process, transaction processing, to name a few, which are outside the scope of this project.

2 Related Work

There is a large body of knowledge on query processing in distributed databases that may provide partial solutions for the above requirements. However the autonomy, distribution and extensible object-oriented data model of the composable mediators architecture proposed here poses new problems different from the ones related to distributed databases. In distributed database systems there is a single site that controls query processing and a centralized catalog, usually replicated in all databases. In contrast to that, in a P2P system there is no central controlling site and all meta-data is distributed among the peers. Various new problems arise from that. Here we mention only some of them: to produce a multidatabase query execution plan (QEP) all peers involved in a query have to cooperate in the compilation process because no peer has complete knowledge of execution cost; peers have to request cost information from other peers over the network which incurs high cost of getting the cost; due to autonomy, cost information may not be available at all; peers can not assume that every other peer is capable of the execution of arbitrary query fragments, therefore predicates might not be freely pushed in the QEP from one mediator to another.

A considerable number of mediator systems [2, 3, 4, 5, 6, 7, 8] have been proposed with varying architectures in terms of their data model, level of distribution and autonomy. Most of the proposed mediator systems have a centralized architecture or fixed two to three tier architecture. Furthermore most of them are based on the relational data model. A distributed relational query processor is proposed in [9], where the focus is on dynamic extensibility and security.

Data management systems based on P2P computing are discussed in [10, 11] where new problems and opportunities arising from the usage of a P2P paradigm are identified. However there is little work on implementation issues of such systems, especially related to large number of cooperating query processors.

Adaptive query processing for single-site query processors has been addressed by various works [12, 13, 14], to name a few. A good overview of adaptive query processing can be found in [15]. Many of the proposed approaches can be integrated with the solution proposed here to implement adaptive behavior of each of the mediator peers. This work is different in that we are interested in the adaptivity of compositions of autonomous mediator peers. A centralized query processor usually has direct access to the data structures of a QEP and therefore it has the full power to modify the QEP at any time and adapt its execution accordingly. In a P2P mediator system a QEP is distributed among all peers participating in the evaluation of a query. Because of autonomy no peer has direct access to the fragments of a global QEP in the other peers. Instead the query processors of autonomous peers have to cooperate through network protocols in order to change a global QEP and adapt during query processing.

A component to extend the optimizer of DB2 DBMS with learning capabilities to repair incorrect statistics and cardinality estimates is described in [16]. Our proposal described in Sect. 5.1 allows to profile any subsystem of the mediator DBMS: query compiler, execution engine and wrappers (and through that the wrapped sources), and access and analyze the profile data in generic way by using the database itself to store the profile data. Multidatabase queries can be used to collect profile data from other mediators and thus distributed mediators can learn from each other.

Compared to other works, our proposal combines a unique set of features: object-oriented data model and query language, extensibility, and distribution and autonomy of the peers.

3 Composable Mediators Architecture

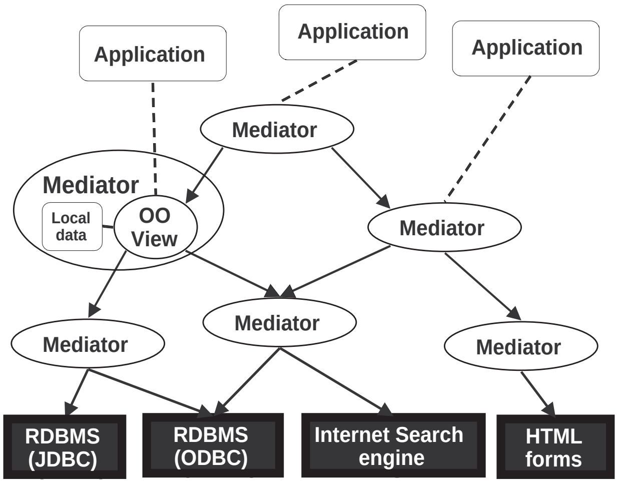


Fig. 1. Logical compositions of mediators

To implement composable mediators we use the P2P mediator system AMOS II [17]. To achieve modularity and distribution each mediator is an autonomous object-relational DBMS with its own query processor, storage and catalog. Knowledge sharing and reuse is based on the capability of the mediator peers to use transparently OO views [18, 19], types and functions from remote mediators or data sources. Logical composition of mediators is achieved when OO views are defined in terms of views, types and functions in other mediators or data sources. The views make the distributed mediators appear to the user as a single virtual database consisting of a number of types (classes) and functions (methods, attributes). Unlike regular OO systems the extents of these types and functions are not explicitly stored in a database but are derived, through an OO multidatabase query language, from data in the underlying data sources and other mediators [20, 21, 19]. We use the term *multidatabase view or function* when a view or function is defined in terms of views, types and functions defined in other mediators.

An example of a mediator composition is shown on Fig. 1. In this example, applications access data stored in several data sources through a collection of composed mediator servers. The arcs connecting the mediator nodes correspond to the relationship 'defined in terms of'. It is important to notice that each of the applications and the mediators may use only some of the distributed mediator definitions. A mediator composition is defined not as static network of mediators but through a query or view definitions together with the multidatabase views involved in that query or view. Thus Fig. 1 is a simplified view of several superimposed mediator compositions.

The modularity and sharability of composable mediators provide additional level of data independence to physical and logical data independence. This level of independence provides:

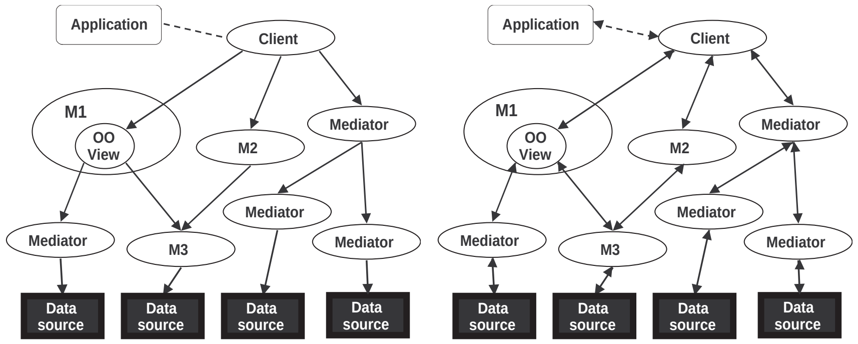
- High level of reuse as knowledge encoded in one mediator can be used by many others.
- Distribution of the mediator modules that reflects the distributed reality where data, knowledge and other resources are usually distributed both organizationally and geographically.
- Preserved autonomy of the data sources, mediators and applications.
- Flexibility to evolve mediators and data sources independently of each other.

Composable mediators can provide a powerful tool to build large-scale data integration systems that are easy to tailor to existing infrastructure, instead of having to adapt the infrastructure to the more rigid approach of centralized integration systems.

4 View Expansion in Composable Mediators

Views are the central concept for data integration in the composable mediators architecture. Therefore the first performance issue we turn our attention to is efficient processing of multidatabase queries over views defined in many peers, in a way that preserves mediator autonomy.

There are two well-studied approaches to implement distributed information systems. The first treats each of the distributed modules of an information system as black boxes. The modules communicate with each other through some protocol without revealing the implementation of the services they export. This is the approach used in CORBA based systems [22]. On the other end are distributed database systems where database views are fully expanded [23] independent of the location of the base tables and views that are used in a view definition. We term the first approach as the *black-box* and the second as *transparent box* approach.



(a) Logical composition of multidatabase views

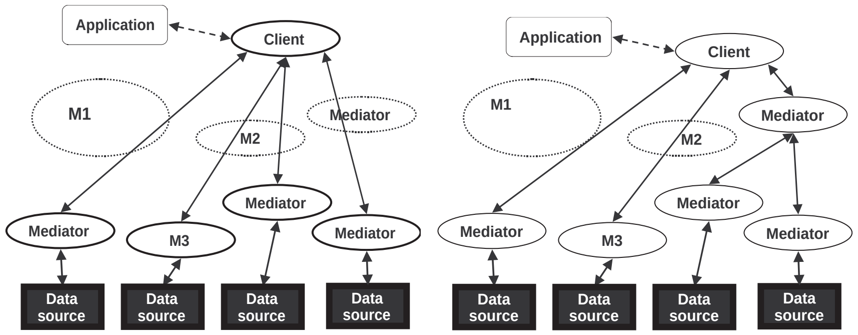
(b) Data flow after no view expansion

Fig. 2. Expansion of multidatabase views

The black-box approach provides full autonomy of the mediators, while at the same time compiling queries without expanding all view definitions may result in sub-optimal execution plans due to missed optimization opportunities and many redundancies in mediator compositions. Figure 2(b) shows the resulting data flow of a QEP for a query compiled against the mediator *Client* from the mediator composition on Fig. 2(a). With no view expansion the mediator *Client* can not 'see' that two of its sub-mediators $M1$ and $M2$ have views implemented in terms of the same common sub-mediator $M3$. As our ongoing experiments show, such redundancies often lead to very inefficient QEPs.

To solve the problems of the black-box approach, DBMS query compilers expand view definitions and merge subqueries. This 'reveals' to the query compiler the information 'hidden' in the view definitions and subqueries which allows for better quality execution plans. Expanding all participating mediator definitions may result in high compilation cost as many more mediators may become 'visible' to the one that compiles a query. Figure 3(a) shows the resulting data flow of a QEP for the mediator composition on Fig. 2(a). Full VE removed the redundant accesses to the mediator $M3$ via $M1$ and $M2$. View expansion may also allow to combine the view definitions at $M1$ and $M2$ and push them down to $M3$. As one may expect, these compilation techniques lead to several orders improvement in the quality of a QEP. On the other hand full VE reveals that the initial three sub-mediators of the mediator *Client* are implemented in terms of four mediators. Therefore the *Client* mediator has to compile a query over a larger number of peers. In large mediator compositions this may lead to prohibitively high compilation cost.

A natural idea is to combine both approaches and treat the mediators as *grey boxes* with varying level of transparency. We have implemented the grey-box approach in a new query compilation technique for P2P mediators, distributed



(a) Data flow after full view expansion

(b) Data flow after partial view expansion

Fig. 3. Expansion of multidatabase views (contd.)

selective view expansion (*DSVE*). In DSVE for better performance mediators can control the level of transparency by selectively expanding only some multidatabase views. To preserve their autonomy mediator peers can decide whether to fulfill or not view definition requests. The performance improvements with DSVE are due to more selective queries, smaller data flows between the servers, fewer servers involved in the query execution while spending relatively little effort in query compilation. An example of the resulting data flow of a QEP produced by DSVE is shown on Figure 3(b) where only the mediator views of $M1$ and $M2$ were expanded.

A pilot implementation of DSVE is described in [24]. Based on this implementation an experimental study with up to 20 mediator peers was performed to determine the effects of selective expansion of multidatabase views on the quality of QEPs. The study shows that one of the most important factors for the overall performance of a P2P mediator system is the topology of the logical mediator composition (i.e. of the graph defined by the mediator peers as graph nodes and the relationship 'defined in terms of' as graph arcs). In our ongoing work we evaluate a heuristic decision procedure for DSVE that utilizes the knowledge of the topology of the logical composition of mediators and targets the view expansion process towards those mediator views that will produce highest increase in QEP quality with the least compilation effort. Our current experiments show that in mediator compositions with 10 and more peers DSVE reduces query compilation time with orders of magnitude with minor losses in the QEP quality.

5 Adaptivity in Mediator Compositions

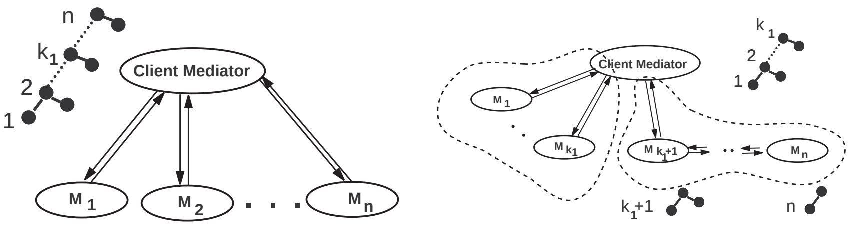
Our current experience from experiments with mediator compositions of over 20 mediators show that incorrect cost and selectivity estimates can lead to orders of magnitude worse query execution plans (QEP). Several factors specific to distributed mediators contribute to the incorrect cost estimates. In most cases it is not possible to acquire statistics about the data stored in the data sources. This is even harder when the data in a source is actually computed and not stored. Imprecise cost modeling may result in that the errors in cost and selectivity estimates increase by orders when propagated through many mediators. Finally data sources, network conditions and mediator load can all change in an unpredictable manner. Therefore it is essential for a mediator system to adapt to an unpredictable and changing environment.

5.1 Integrated Self-Profiling

As a basis for adaptivity, a mediator system should be able to measure various parameters of its environment and its own operation, store this measurements and use them to detect sub-optimality and to adapt by recomputing the affected QEPs.

Our approach to measure system performance and manage measurement data is to integrate a database-based profiling system with the query processor of each mediator peer. That will enable the query processor of a mediator to measure parameters related to its own operation, the sources it accesses and the network, and then use the accumulated information for better future decisions. The main idea behind this integrated profiling approach is to use the mediator system itself in a reflective manner and store all measurement data in the database itself. The benefits of this approach are that the full power of an OO query language will be available to update, retrieve and analyze the distributed measurement data. Potentially there may be large amount of profile data with dynamically changing distribution across many mediator peers. Using the multidatabase query capabilities of the mediator system in a reflective manner to access the profile data would allow to let the system automatically compute the best access path to the data without the need to hard-code it and to easily modify the decision-making procedures inside the optimizer.

Because AMOS II is a main-memory DBMS, we can expect very fast updates and retrievals of the measurement data. This will allow to minimize the the performance penalty of profiling during normal system usage. The extensibility of AMOS II allows to define custom data structures and functions to store and update profiling data in the most efficient manner while still preserving a query interface to that data. Finally the architecture of the AMOS II mediator system allows any system component to be profiled in a generic manner. An interesting direction is to profile the operation of all critical components of the query engine and to introduce adaptivity not only at the level of the query execution plans but other system components as well, e.g. the query compiler itself.



(a) Centralized QEP

(b) Decentralized QEP

Fig. 4. Rebalancing of a multidatabase QEP

The major challenges are how to minimize the performance penalty of profiling, to ensure that the necessary profiling data can be accessed very fast as this will be done from inside the query engine and finally the ability to dynamically control what parameters are being measured.

5.2 Adaptive Rebalancing of Multidatabase QEPs

As a first application of the integrated self-profiling this project will focus on adapting the distributed data flow of the multidatabase QEPs in the AMOS II system.

In [21] we investigated rebalancing of multidatabase QEPs that allows the query compiler to generate decentralized plans at each mediator. QEP rebalancing takes a centralized plan (Fig. 4(a)) where all communication between one mediator and all its direct sub-mediators passes through the controlling mediator and transforms it whenever favorable into a plan with side-wise information passing, where some of the communication is performed directly between the sub-mediators (Fig. 4(b)). For this sub-plans of the centralized QEP are sent to the nearest mediators (in terms of logical composition) and further compilation of the sub-plans is delegated to neighbor peers. The peers in turn may further decide to apply rebalancing to the sub-plans received for compilation.

While [21] shows that distributed QEP rebalancing removes some of the overhead of logical mediator composition, this is done in a static manner. Future work for this project is to extend QEP tree rebalancing to allow mediators to automatically adapt the data flow of distributed QEPs to changes that may occur in a P2P mediator system.

Important research issues related to adaptive QEP rebalancing, and to adaptivity in general are: detecting sub-optimal performance and adapting to it; reuse parts of a QEP when re-adapting to save compilation work; reuse of the intermediate query execution results - if only some of mediators' plans are reoptimized only the execution of a sub-plan could be restarted instead of recomputing the whole result from scratch.

References

[1] Wiederhold, G.: Mediators in the architecture of future information systems. IEEE Computer **25** (1992) 38–49

[2] Haas, L.M., Kossmann, D., Wimmers, E.L., Yang, J.: Optimizing queries across diverse data sources. In Jarke, M., Carey, M.J., Dittrich, K.R., Lochovsky, F.H., Loucopoulos, P., Jeusfeld, M.A., eds.: Proceedings of 23rd International Conference on Very Large Data Bases, VLDB'97, Athens, Greece, Morgan Kaufmann (1997) 276–285

[3] Tomasic, A., Raschid, L., Valduriez, P.: Scaling access to heterogeneous data sources with disco. IEEE Transactions on Knowledge and Data Engineering **10** (1998) 808–823

[4] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J.D., Vassalos, V., Widom, J.: The tsimmis approach to mediation: Data models and languages. Journal of Intelligent Information Systems (JIIS) **8** (1997) 117–132

[5] Richine, K.: Distributed query scheduling in diom. Tech. report TR97-03, Computer Science Dept., University of Alberta (1997)

[6] Du, W., Shan, M.: Query processing in pegasus. In Bukhres, O.A., Elmagarmid, A., eds.: Object-Oriented Multidatabase Systems: A Solution for Advanced Applications. Pretince Hall, Englewood Cliffs (1996)

[7] Haas, L.M., Schwarz, P.M., Kodali, P., Kotlar, E., Swope, J.E.R.W.C.: Discoverylink: A system for integrated access to life sciences data sources. IBM Systems Journal **40** (2001) 489–511

[8] Liu, L., Pu, C.: An adaptive object-oriented approach to integration and access of heterogeneous information sources. Distributed and Parallel Databases **5** (1997) 167–205

[9] Braumandl, R., Keidl, M., Kemper, A., Kossmann, D., Kreutz, A., Seltzsam, S., Stocker, K.: Objectglobe: Ubiquitous query processing on the internet. VLDB Journal **10** (2001) 48–71

[10] Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I.: Data management for peer-to-peer computing: A vision. In: Workshop on the Web and Databases, WebDB 2002, Madison, Wisconsin (2002) SIGMOD 2002.

[11] Gribble, S., Halevy, A., Ives, Z., Rodrig, M., Suciu, D.: What can databases do for peer-to-peer? In: WebDB Workshop on Databases and the Web. (2001)

[12] Urhan, T., Franklin, M.J., Amsaleg, L.: Cost based query scrambling for initial delays. In Haas, L.M., Tiwary, A., eds.: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 1998, Seattle, Washington, USA, ACM Press (1998) 130–141

[13] Kabra, N., DeWitt, D.J.: Efficient mid-query re-optimization of sub-optimal query execution plans. In Haas, L.M., Tiwary, A., eds.: Proceedings of the ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, ACM Press (1998) 106–117

[14] Avnur, R., Hellerstein, J.M.: Eddies: continuously adaptive query processing. ACM SIGMOD Record **29** (2000) 261–272

[15] Hellerstein, J.M., Franklin, M.J., Chandrasekaran, S., Deshpande, A., Hildrum, K., Madden, S., Raman, V., Shah, M.A.: Adaptive query processing: Technology in evolution. IEEE Data Engineering Bulletin **23** (2000) 7–18

[16] Stillger, M., Lohman, G.M., Markl, V., Kandil, M.: Leo - db2's learning optimizer. In Apers, P.M.G., Atzeni, P., Ceri, S., Paraboschi, S., Ramamohanarao, K., Snodgrass, R.T., eds.: Proceedings of 27th International Conference on Very Large Data Bases, Roma, Italy, Morgan Kaufmann (2001) 19–28

[17] Risch, T., Josifovski, V.: Distributed data integration by object-oriented mediator servers. Concurrency and Computation: Practice and Experience **13** (2001) 933–953

[18] Josifovski, V., Risch, T.: Functional query optimization over object-oriented views for data integration. Journal of Intelligent Information Systems **12** (1999) 165–190

[19] Josifovski, V., Risch, T.: Integrating heterogenous overlapping databases through object-oriented transformations. In Atkinson, M.P., Orlowska, M.E., Valduriez, P., Zdonik, S.B., Brodie, M.L., eds.: Proceedings of 25th International Conference on Very Large Data Bases, VLDB'99, Edinburgh, Scotland, UK, Morgan Kaufmann (1999) 435–446

[20] Fahl, G., Risch, T.: Query processing over object views of relational data. VLDB Journal **6** (1997) 261–281

[21] Josifovski, V., Katchaounov, T., Risch, T.: Optimizing queries in distributed and composable mediators. In: Proceedings of the Fourth IFCIS International Conference on Cooperative Information Systems, CoopIS'99, Edinburgh, Scotland, IEEE Computer Society (1999) 291–302

[22] Soley, R., Stone, C., eds.: Object Management Architecture. John Wiley & Sons, New York (1995)

[23] Özsu, M.T., Valduriez, P.: Principles of Distributed Database Systems. Second edition edn. Prentice Hall (1999)

[24] Katchaounov, T., Josifovski, V., Risch, T.: Distributed view expansion in composable mediators. In Etzion, O., Scheuermann, P., eds.: Proceedings of the 7th International Conference on Cooperative Information Systems, CoopIS 2000. Volume 1901 of Lecture Notes in Computer Science., Eilat, Israel, Springer (2000) 144–149