Supporting Autonomy for Information Systems in a Changing Environment

J. Kusch and G. Saake

Institut für Technische Informationssysteme, Abteilung Datenbanken,
Otto-von-Guericke-Universität Magdeburg, D–39106 Magdeburg, Germany,
E-Mail: {kusch|saake}@iti.cs.uni-magdeburg.de

In the development of information systems, object-oriented specification (e.g. TROLL [4]) is useful for conceptually modeling of the universe of discourse. Viewing an information system as a collection of communicating objects is close to the intuitive perception of such systems on a conceptual level. Nowadays complex information systems (e.g. knowledge bases) have to be flexible to accommodate for changes in organization structures. Mandatory features are *distribution* for decentralization support and *scalability* for modular system increase. Furtheron processing in parallel enables an important *speedup*.

In general, distributed database systems [7, 6] have to deal with a lot of *tradeoffs*, e.g. data replication versus data transfer, reuse versus autonomy, transparency versus efficiency. Our goals in gaining distribution for an object-oriented data model are: (1) Execution autonomy: For *decentralization* support, federation does neither interfere with local operations nor is any knowledge of the federation needed for performing local operations. (2) Transparency aspects: For *scalability* support transparency of location and migration enables objects to be used without knowledge of their location and movement of objects within a system without affecting the operations. In conjunction with horizontal and vertical class partitioning, the system could be expanded in scale without changing the specification or the references. (3) Access parallelism: For *speedup* support, access time to partitioned classes has to be decreased by parallelized mechanisms.

To gain this kind of autonomy, transparency and parallelism for object-oriented specification, at least the following problems have to be achieved: Object-oriented specification consists of a global abstract description as conceptual model without non-functional requirements (e.g. distribution, persistence, exception handling). A distributed infrastructure consists of a set of inter-connected loosely coupled nodes with own processors and disk-spaces. *On which 'level' distribution has to be combined with the conceptual model?* A complex information system structure, which is described as conceptual entirety, is transparently distributed over several nodes. Important is a partial use of the information system, although not all nodes are available or reachable due to a side failure. *How to achieve a maximum of node autonomy for a distributed system (C. J. Dates first (of 12) rule for distributed database systems [2]) supported by the distributed schema catalog?*

Existing approaches in the area of database systems mostly do not pay enough attention to the opportunities of autonomy and scalability, which are getting increasingly important by a new generation of *parallel hardware clusters* [3].

This position paper emphasizes on the *structural aspects* of the object model. Mandatory structural features of object-oriented database systems and information systems as pointed out in i.e. [4, 1] are *object identity*, *types and classes*, *type and class hierarchies*, *complex objects* and *views*. Operationalizing this structures, we propose three levels on which autonomy has to be considered:

Operational level: Due to the increase of hardware performance, the schema catalog of an object-oriented database system has to perform the mapping from the conceptual specification structure to the distributed operational structure. Our approach proposes as architecture of the distributed schema catalog a *replicated location schema* with distributed local schemas. This allows to perform consistency preserving operations even if nodes in the context of this operations are offline. So, node autonomy is achieved in spite of complete distribution transparency. For orthogonally influencing distribution, a meta-schema is introduced.

Conceptual level: The enhancement of the object model with integrity constraints represents a tradeoff to autonomy. Thus the use of integrity constraints has to be performed preconceived with the view to the needed autonomy. This could be labeled as autonomy design phase.

Design phases level: To gain execution autonomy for an information system, on base of the above mentioned operational level and conceptual level the phases for database design could be stated as: (1) Requirements acquisition. (2) Autonomy design. (3) Conceptual modeling. (4) Implementation: Conceptual model as DDL of the distributed schema catalog. (5) Distribution design: *Late phase* of the design process for *tuning* distribution issues.

Collectively, the presented *syntax independent* work depicts fundamentals for specification language concepts and their operationalization, when dealing with distribution. The disadvantages of transparency aspects within our approach could be improved by dynamic query optimization. *Query objects* [5] with internal knowledge of the state of the infrastructure are a rudiment for this problem.

References

1. M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, and S. Zdonik. The Object-Oriented Database System Manifesto. In W. Kim, J.-M. Nicolas, and S. Nishio, editors, *DOOD*, pages 223–240. North Holland, 1990.
2. C. J. Date. *An Introduction to Database Systems*. Addison-Wesley, 1990.
3. J. Gray. Super-Servers: Commodity Computer Clusters Pose a Software Challenge. In G. Lausen, editor, *BTW95*, pages 30–47. Springer-Verlag, 1995.
4. R. Jungclaus, G. Saake, T. Hartmann, and C. Sernadas. TROLL – A Language for Object-Oriented Specification of Information Systems. *ACM Transactions on Information Systems*, 1995. *To appear.*
5. R. Jungclaus, G. Saake, and C. Sernadas. Using Active Objects for Query Processing. In R. Meersman, W. Kent, and S. Khosla, editors, *Proc. of the 4th IFIP WG 2.6 Conf. DS-4, Windermere (UK), 1990)*, pages 285–304. North-Holland, 1991.
6. M. T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, 1991.
7. E. Rahm. *Mehrrechner-Datenbanksysteme*. Addison-Wesley, 1994.

This article was processed using the $\LaTeX$ macro package with LLNCS style