Research Issues in Large Workflow Management Systems

Gustavo Alonso Hans-Jörg Schek
Database Research Group
Institute for Information Systems
ETH Zentrum, Zürich CH-8092, Switzerland
E-mail: {alonso,schek}@inf.ethz.ch

April 30, 1996

Abstract

In this position paper we describe what we believe are fundamental weaknesses of existing commercial workflow products and how database technology can be used to address these issues. By *exporting* database technology beyond the transactional domain, we expect to be able to design scalable, robust, failure-resilient and flexible workflow systems to be used in application domains ranging from business process to computer integrated manufacturing.

1 Introduction

Workflow Management Systems are quickly becoming the technology of choice to implement large and heterogeneous distributed execution environments where sets of interrelated tasks can be carried out in an efficient and closely supervised fashion. In many ways, a workflow management system is not different from a sophisticated scheduler in which the scheduling is performed based on task dependencies, organizational structure, staff availability, and existing computing infrastructure. It is precisely this characteristic that makes workflow management systems so appealing, as it makes them match very well current organizational and technological trends: decentralization of the corporation, decentralization of the decision making, the need for very detailed information about every day activities, and the increasing availability of distributed processing technology (WWW, CORBA, OLE, Java).

Proof of the ever increasing interest in workflow management is the large number of commercial products that have appeared in the last few years. These products are finding an eager market and opening up important research opportunities. In particular, there are many instances in which the expectations from the users and the actual features provided by the systems are not well correlated. The main reason for this is that the requirements of a workflow system in terms of scalability and system wide reliability exceed those of database and transaction processing technology [Gaw94], which is the current state of the art. As a consequence, workflow technology is far from being as mature as existing transaction processing and database technology and lacks the performace, fault-tolerance, and flexibility that is provided by today's databases. The obvious challenge is to extend existing techniques and develop new ones to make workflow management the backbone of the information processing technology of the enterprise. The not so obvious question is which techniques are useful in a workflow environment. In this position paper we outline what we believe is the relevant technology that can be used to enhance the functionality of workflow management systems. We motivate our ideas by pointing out the weaknesses in existing commercial products and briefly describing how databases and transaction monitor technology can serve as the basis to address these problems.

2 Limitations of Existing Systems

In the workflow area, and in spite of the many research projects underway, the state of the art has so far been determined by commercial products. Hence, to understand the state of the art, it is important to understand where workflow comes from. The origins of commercial workflow systems can be traced back to office automation, image processing or computer supported cooperative work [Hol94]. The emphasis in these environments has been *routing*, *sharing* and *cooperation*. Issues such as performance, scalability or reliability are hardly ever considered by this type of systems [KAGM96] and these characteristics have been inherited by workflow products. There are no commercial workflow products based on OLTP (On-Line Transaction Processing) or database technology, although many of them use databases as the underlying repository and some incorporate ideas that can be related to functionality found in commercial transaction monitors. As a consequence, the robustness and technological maturity reached in the database area is all but lacking in workflow systems [Gaw94].

Nowadays there several hundred commercial products that claim to be workflow tools. Of these, only a handful are true workflow engines. Some of the most relevant systems in the market include: **Action-Workflow System**, of Action Technologies; IBM's **FlowMark**; **WorFlo Business Systems** of FileNet; **InConcert**, produced by XSoft, a division of Xerox Corp; **OmniDesk**, of Sigma Imaging Systems Inc.; **ProcessIT**, of AT&T Global Information Solutions (formerly NCR); **Staffware**, of Staffware Corporation; **Regatta**, of Fujitsu, currently incorporated into ICL's TeamWARE; and **OPEN/workflow**, a WANG's product. It is also important to mention that there are a multitude of other products being developed as third party applications on top of distributed platforms such as LotusNotes. Such products play a role similar to the many third party tools used to interface with a database management system (SQL forms, for instance) and should not be considered as workflow engines.

As has been pointed out many times in the literature [GHS95], the first generation of workflow engines is started to find wide acceptance, but available products are far from providing optimal solutions. The following are some of their most important limitations:

- **Interoperability**: existing systems are almost totally incompatible. The situation is similar to that of databases before the widespread acceptance of the relational model. In spite of efforts like those of the Workflow Management Coalition [Hol94], current products incorporate in the design very concrete and exclusive interpretations of the world (ActionWorkflow being the best example) making practically impossible to federate different systems. These incompatibilities are not just the syntax or the platform, but the very interpretation of workflow execution. Some other systems are so tied to the underlying support system that is unfeasible to extend their functionality to accomodate other workflow interpretations (for instace, workflow tools developed on top of Lotus Notes).

- **Scalability**: as cooperative tools, existing workflow engines have been designed with small groups in mind. Only afterwards, when users realized the potential of workflows, these engines have been applied to large scale environments. However, the inherent restrictions in the designs prevent existing systems from being able to cope with a fraction of the expected load, which can be as high as several hundred thousand processes and tens of thousands of users [KAGM96].
- **Availability**: the degree of resilience to failures of current systems is almost null. Current products have a single point of failure (the database) and no mechanism for backup or efficient recovery. This is not as much a flaw as a design decision, since these products were initially intended for small groups and small loads.
- **Correctness** of execution of several workflows has not been considered an issue so far. As long as we deal with loosely coupled systems where no integrity constraints exist that span multiple systems, the single steps of a workflow can be executed without any further control. If, however, in a more tightly coupled system dependencies must be observed, uncontrolled interferences of several parallel workflows could be prevented with similar techniques to those used in database transactions.
- **Architecture**: the most common workflow design nowadays is a single database on top of which a single workflow engine provides services to several clients. These architectures quickly become a bottleneck. Some enhanced architectures have been proposed [BMR94, AKA+94, AAE+95], but so far they still remain research ideas.

In general, current workflow products cannot be used as the basis for an enterprise wide system. The basic elements of today's information technology within a corporation are databases and on-line transaction processing. In many ways, the state of the art of commercial workflow systems is far behind that and yet they are being presented as tools to handle environments much more complex and demanding than corporate databases and transaction processing.

3 Databases and Workflows

Many of the weaknesses described above are still to be addressed and, in some cases, even to be recognized as problems [GHS95, KAGM96]. However, there is a solid base of knowledge that can be used to solve many of these issues. Most of it involves extending database technology to the new application area. In some cases it is not a straightforward extension since workflows have greater demands than databases, but work in this area will both benefit workflow managament systems and databases [Gaw94]. As a first step, we suggest that these extensions should target the weaknesses discussed in the previous section:

- **Flexibility**: without advocating a standard such as SQL (workflow technology is still too far from reaching such a point), it is very important that workflow systems be designed independently of the characteristics of the workflow specification language (Petri-nets, state charts, transactional dependencies, et cetera). A workflow engine must be flexible enough to execute any type of workflow relationship regardless of the way it is specified. To achieve this, it is necessary to develop a much better understanding of the execution model of workflows. Current research is still too bound to transaction models, which is very limiting when defining generic workflows [AAE+96].

- **Interoperability**: database federation, schema integration and data warehousing are all relevant topics in the database community. Exactly the same problems will exist with workflow engines as different parts of an organization develop their own workflow environments which, later, will have to be integrated. Workflow architectures need to be designed with this in mind. It does not suffice to have a common model (like the one proposed by the Workflow Managament Coalition [Hol94]), as the problem involves much more complicated issues. In this area, technology like CORBA or OLE may have a considerable impact [MSKW96].

- **Availability**: very large workflow management systems will involve several thousand users, hundreds of thousands of concurrently running processes and several thousand sites distributed over wide area networks. These are actual requirements on commercial systems [KAGM96]. In these environments continuous availability is crucial, in the same way that continuous availability is the key to many banking and corporate database applications. Database backup and replication techniques could be used to automatically transferred the load of a failed component to other servers in the system without requiring to stop ongoing executions. Moreover, this also implies a built-in mechanism for load balancing, redirecting requests to other servers when excessive load may affect the quality of service provided by a server.

- **Concurrency**: Correctness of parallel transactions in databases is ensured by serializabililty. This is a too narrow definition of correctness, not suitable for workflows environments because workflows span multiple systems and their execution may take several days or weeks. For similar reasons, the 2PC protocol, standard for distributed transactions and the basis for transaction processing monitors, may not always be suitable for workflows. If applied, subsystems could be blocked for long periods of time. Instead, open nested transactions could be used to commit subtransactions early and avoid blocking due to low level conflicts if the higher-level operations do not conflict. This approach separates the responsibilities: The correctness of single steps is under the responsibility of the subsystem that executes the step while the correctness of the whole workflow is delegated to the more abstract workflow level where the information is available what interleavings should not be allowed. [BDS+93] has proposed to combine this technology with workflow specification. In [DSW94, SSW95] these ideas are used as an alternative for the 2PC protocol in distributed transaction processing.

- **Exception Handling**: In workflow environments, the system cannot arbitrarily abort an operation. If an exception arises, human interaction is often required to resolve it, as it is generally unwise to expect the workflow designer to be able to predict all possible exceptions in the execution of a process. In many cases, given the nature and characteristics of the applications, these exceptions are unpredictable and range from simple database-like conflicts to changes in personal, changes in legislation, and economic adjustments. This is a departure from standard database transaction technology since the system does not have the knowledge to resolve all possible exceptions. The proper level of abstraction to resolve this problem is the workflow level where at least a notification mechanisms must be included. Similar issue arise in the coordination of intersystem dependencies of CIM subsystems [NSSW94] and some of the ideas developed in that are could be applied in workflows. Moreover, the problem of having a recoverable state for long computations, fundamental to implement a flexible exception handling mechanism, is also started to be addressed in the transaction processing ares[ST96].

- **Fault-tolerance**: in large scale workflow systems there will be a variety of components involved in the execution of every activity within a process. The system must tolerate failures of any of these components. For instance, when a workflow engine is not available, it should be possible to automatically connect the clients to other engines to continue the execution of the process. Depending on the type of failure, the worst scenario should be that a process execution is interrupted until the failure is repaired. Upon recovery, execution should be resumed at the same point where it was interrupted. Some work has been done in this area [BMR94, AKA+94, AAE+95], but much remains to be done.

- **Performance**: existing systems do not consider performance to be an issue. But it is possible that all or a great part of a workflow process is solely comprised of automatic tasks (with no human intervention) in which case the workflow engine becomes the major source of delays in the execution of a process. While there is a trade off between the flexibility provided by workflow systems (execution not based on compiled code) and the efficiency of conventional systems (like TP-monitors where the execution is based on a compiled program), the system should be able to provide mechanisms to enhance the performance when automatic tasks are involved. The need for high performance distributed processing is not exclusive of transaction monitors [Gaw94].

- **Scalability**: the major drawback of existing systems is that they are not able to cope with large scale environments. In the best possible scenarios, commercial systems can be used with up to 40 users and a few hundred processes running concurrently. This is far from the figures one encounters as common requirements. To tackle realistic applications, the sytem designed should be easily scalable to any number of users and concurrent processes. This implies that replication techniques must be incorporated within the architecture and all components must be designed in a modular fashion to allow customization of the system.

4 Conclusions

While databases guarantee the safe storage and easy acess to massive amounts of data, workflow management systems are intended as the basic support for information flow in those same environments where databases are used. However, the current state of the art of workflow engines is far behind that of databases and transaction processing. The successful incorporation of workflow systems into the corporation greatly depends on the ability to make workflows a technology as mature and resilient as existing databases. In this position paper we have described what, in our opinion, are important weaknesses of commercial products. Our approach to these problems is to extend database technology to the new application environment without necessarily seeing workflows as a database application. Workflow management systems can benefit from existing technology, but also require new solutions as their demands are considerable larger than those of today's databases.

References

[AAE+95] G. Alonso, D. Agrawal, A. El Abbadi, C. Mohan, R. Günthör, and M. Kamath. Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management. In *IFIP WG8.1 Working Conference on Information System Development for Decentralised Organizations*, Trondheim, Norway, August 1995.

[AAE+96] G. Alonso, D. Agrawal, A. El Abbadi, M. Kamath, R. Günthör, and C. Mohan. Advanced Transaction Models in Workflow Contexts. In *Proceedings of the 12th International Conference on Data Engineering*, New Orleans, Louisiana, USA, February 1996.

[AKA+94] G. Alonso, M. Kamath, D. Agrawal, A. El Abbadi, R. Günthör, and C. Mohan. Failure Handling in Large Scale Workflow Management Systems. Research Report RJ 9913, IBM Almaden Research Center, November 1994.

[BDS+93] Y Breitbart, A. Deacon, H.-J. Schek, A. Sheth, and G. Weikum. Merging Application-centric and Data-centric Approaches to Support Transaction-oriented Multi-system Workflows. *ACM Sigmod Record*, September 1993.

[BMR94] D. Barbara, S. Mehrota, and M. Rusinkiewicz. INCAS: A Computation Model for Dynamic Workflows in Autonomous Distributed Environments. Technical report, Matsushita Information Technology Laboratory, April 1994.

[DSW94] A. Deacon, H.-J. Schek, and G. Weikum. Semantics-based Multilevel Transaction Management in Federayed Systems. In *Proceedings of the 10th International Conference of Data Engineering*, Houston, Texas, USA, February 1994.

[Gaw94] D. Gawlick. High Performance TP-Monitors – Do We Still Need to Develop Them? *IEEE Bulletin of the Technical Committee on Data Engineering*, 17(1):16–21, March 1994.

[GHS95] D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases*, 3(2):119–153, April 1995.

[Hol94] D. Hollinsworth. The workflow reference model. Technical Report TC00-1003, Workflow Management Coalition, December 1994. Accessible via: http://www.aiai.ed.ac.uk/WfMC/.

[KAGM96] M. Kamath, G. Alonso, G. Günthör, and C. Mohan. Providing High Availability in Very Large Worklflow Management Systems. In *In Proceedings of the Fifth International Conference on Extending Database Technology (EDBT'96)* , Avignon, France, March 1996. Also available as IBM Research Report RJ9967, IBM Almaden Research Center, July 1995.

[MSKW96] J.A. Miller, A. Sheth, K.J. Kochut, and X. Wang. Corba-based run-time architectures for workflow management systems. *Journal of database Management*, 7, 1996.

[NSSW94] M.C. Norrie, W. Schaad, H.-J. Shek, and M. Wunderli. Exploiting Multidatabase Technology for CIM. Technical report, Computer Science Department, Database Research Group, July 1994.

[SSW95] W. Schaad, H.-J. Schek, and G. Weikum. Implementation and Performance of Multi-Level Transaction Management in a Multidatabase Environment. In *Proceedings of the 5th International Workshop on research Issues on Data Engineering*, Taipei, Taiwan, March 1995.

[ST96] B. Salzberg and D. Tombroff. DSDT: Durable Scripts Containing Database Transactions. In *Proceedings of the 12th International Conference on Data Engineering*, New Orleans, Louisiana, USA, February 1996.