8-15-1997

# Supporting Telecom Business Processes by Means of Workflow Management and Federated Databases

Wim Nijenhuis
*KPN Research, The Netherlands*, r.w.nijenhuis@research.kpn.com

Willem Jonker
*KPN Research, The Netherlands*, w.jonker@research.kpn.com

Paul Grefen
*University of Twente, The Netherlands*, grefen@cs.utwente.nl

# Supporting Telecom Business Processes by Means of Workflow Management and Federated Databases

**Wim Nijenhuis** and **Willem Jonker**
KPN Research, The Netherlands
{r.w.nijenhuis,w.jonker}@research.kpn.com

**Paul Grefen**
University of Twente, The Netherlands
grefen@cs.utwente.nl

## Introduction

Core telecom business processes such as network planning, network management, and service management operate on large and heterogeneous collections of autonomous information systems. As requirements to these systems have changed over time, they have been adapted frequently and consequently they serve quite different demands than those they were originally designed for. The systems have become large monoliths that are difficult to maintain and have lost the flexibility to support changing business processes effectively.

Both the distributed systems community and the information systems community work on the development of architectures, techniques, and tools to address these problems. For instance, TINA (Telecommunication Information Networking Architecture) defines an open software architecture for telecommunication services. The TINA Distributed Processing Environment separates the functional applications and physical infrastructure and thus enables the use of heterogeneous systems. For application developers, it is often difficult to understand how these architectures, techniques, and tools can be brought together to solve their problems. We have come across this problem several times inside Dutch telecom (PTT). The main problem is how to use new technologies such as workflow management and federated databases in an overall information system architecture.

Within PTT, the overall system architecture is based on an architecture similar to the OSCA architecture [10]. Therefore, we present our approach on integrating workflow management and federated databases in the context of OSCA. In the extended architecture, monolithic information systems as such disappear and business processes are supported by workflows. This paper discusses the basic aspects of the architecture and its benefits.

## Related work

Several approaches to flexible information system architectures have been described. An important class is formed by approaches based on the concept of building blocks that provide 'chunks' of encapsulated functionality with clear interfaces. A recent example of this class is an approach towards structuring telecommunication systems [9]. A second important class is formed by approaches based on layering the functionality of systems. The best-known examples are probably the ISO-OSI reference model for network systems and the ANSI-SPARC three-schema architecture for database systems. In information system architectures, middleware layers have been proposed to act between information servers and user clients, e.g. [12]. We combine the building block and layering approaches in our approach to architectures.

In the field of federated database systems, a number of relevant system aspects have been investigated recently. Architectures for multidatabase systems have been described allowing for the integration of multiple databases [11]. Traditional basic transaction mechanisms have been extended to deal with the characteristics of federated systems, e.g. transactional protocols for integrity checking [8]. In the field of workflow management systems, attention has been paid to reference architectures. A well-known architecture is that of the Workflow Management Coalition [13]. The work in [7] proposes a more detailed reference architecture allowing for modular extension. Advanced transaction models [4, 6, 5] are important ingredients to obtain transactional workflows [1].

## A layered architecture

The OSCA architecture [10] is well known within the Telecom industry. The key point of OSCA is to distinguish three layers in the architecture of information systems: the data layer, the processing layer and the user layer (see Figure 1). Each layer consists of building blocks. A building block is an encapsulated set of computer programs and data schema's, providing its functions to other building blocks through explicit interfaces. Interfaces are specified using an unambiguous definition language, e.g. CORBA-IDL. With the separation in layers, the architecture is based on separation of concerns, enabling reuse of common data and common functionality.

The *data layer* groups the corporate data management functionality. This layer has the most stable structure over time. The corporate data is strongly related to the high-level business strategy and has a company wide scope. Changes in the architecture of the data layer are typically related to business changes (new product, services, types of customers). Note that there is private data in all three layers, which is owned by a building block. The *processing layer* contains business functions and management functionality, based on the enterprise model. Business functions executed by the processing layer usually require access to corporate data in the data layer. Changes in the processing layer are typically related to a fundamentally different way of doing the same business, often related to business process reengineering. The *user layer* contains the human interaction functionality, based on the human business tasks and goals. It serves as an agent for carrying out the user tasks by accessing functionality of other building blocks. The user can only access business functionality or corporate data via the user layer. Changes in the way users perform their tasks result in changes in the user layer. As this may happen frequently, flexibility is of great importance to the user layer.

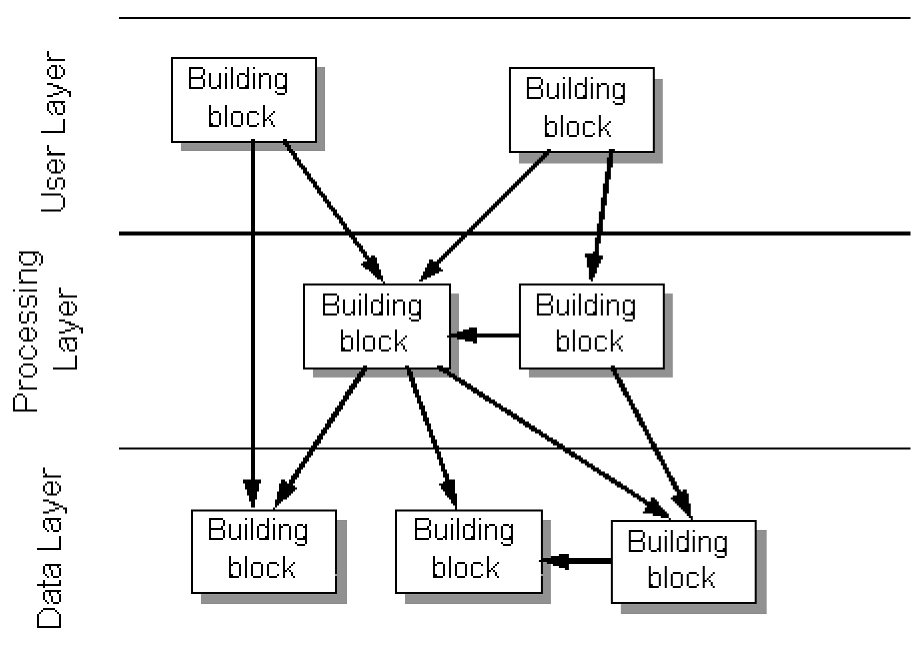


***Figure 1: The OSCA Architecture.***

Below, we show how we integrate federated databases and workflow management systems to extend the basic OSCA architecture to cope with complex environments.

## Positioning federated databases

By separating data from applications, OSCA makes corporate data available to other applications. The question to be answered then is how to access these corporate data, i.e. how do user and processing layer building blocks access data layer building blocks? In order to obtain a maintainable architecture, it is preferred to have uniform access to building blocks. A popular way to obtain uniform access to databases is to use gateway technology. This technology only deals with syntactic uniformity and fails to address the problem of semantic heterogeneity of the data layer building blocks.

The field of federated databases addresses the problem of uniform access to collections of heterogeneous databases. A federated database management system (FDBMS) is a 'virtual' DBMS on top of a heterogeneous collection of databases. It provides support for schema integration to deal with syntactic and semantic heterogeneity, transaction management over multiple autonomous DBMSs, and query optimization over multiple DBMSs. Research into federated databases was originally motivated by the problem of legacy systems [2, 11]. However, we argue that database federation is a good structuring principle on its own, useful for structuring newly developed information systems.

In the OSCA architecture, we position a FDBMS in the data layer as a uniform mechanism for accessing the data layer building blocks, i.e. we force the higher level building blocks to access corporate databases through the FDBMS (see the bottom part of Figure 2). This simplifies the implementation of building blocks in the processing layer and user layer. The FDBMS makes it easier to offer a database schema that is tuned to the needs of a user layer or processing layer building block, to coordinate transactions over multiple data layer building blocks, and to improve performance of access to multiple data layer building blocks.

## Positioning WFM systems

We position workflow management systems based on the reference architecture in [7]. In this architecture, workflow client, workflow server, and database interface functionality are clearly distinguished. Coupling with (legacy) applications is positioned at the server level (usually batch applications) or at the client level (usually interactive applications). Coupling between multiple workflow management systems is positioned at the server level. Coupling with the underlying multidatabase can be structured according to the various data sets distinguished in [7]: organization data, product data, schema data, process data, management data, and application data.

When relating workflow systems to the OSCA-architecture, they can be seen as the means to specify and control the use of business applications (i.e. processing layer building blocks) and corporate data (i.e. data layer building blocks). Thus WFM offers a way to specify the rules related accessing different business applications and corporate data. In current IT systems, business rules are either implemented separately for every application or left to the responsibility of the user. Positioning WFM in the OSCA architecture enables locating business rules within the architecture.

We position WFM in the processing layer as a uniform access mechanism for accessing the processing layer building blocks, i.e. we force the user layer building blocks to access applications through the WFM engine (see the top part of Figure 2). Thus a WFM engine enhances the architecture by enabling an implementation for the other building blocks that is more or less independent of a specific business process. The WFM engine coordinates the use of business functions implemented by processing layer building blocks even if they are on heterogeneous systems. This simplifies the implementation of building blocks in the user layer.

## Business process transactions

The architecture we propose is a layered architecture consisting of a workflow management system on top of a federated database system. An important issue is how to support transactions in such an environment. Both the database community and the workflow community work on new transaction models that can support advanced applications [1,5]. For our architecture we adopt a two-level transaction model: a basic transaction model supported by the federated database management system and an advanced transaction model supported by the workflow manager. The basic transaction model supports the traditional ACID properties, e.g. a 2PC/2PL or nested transaction model. The advanced transaction supports typical long-lived business process transactions.

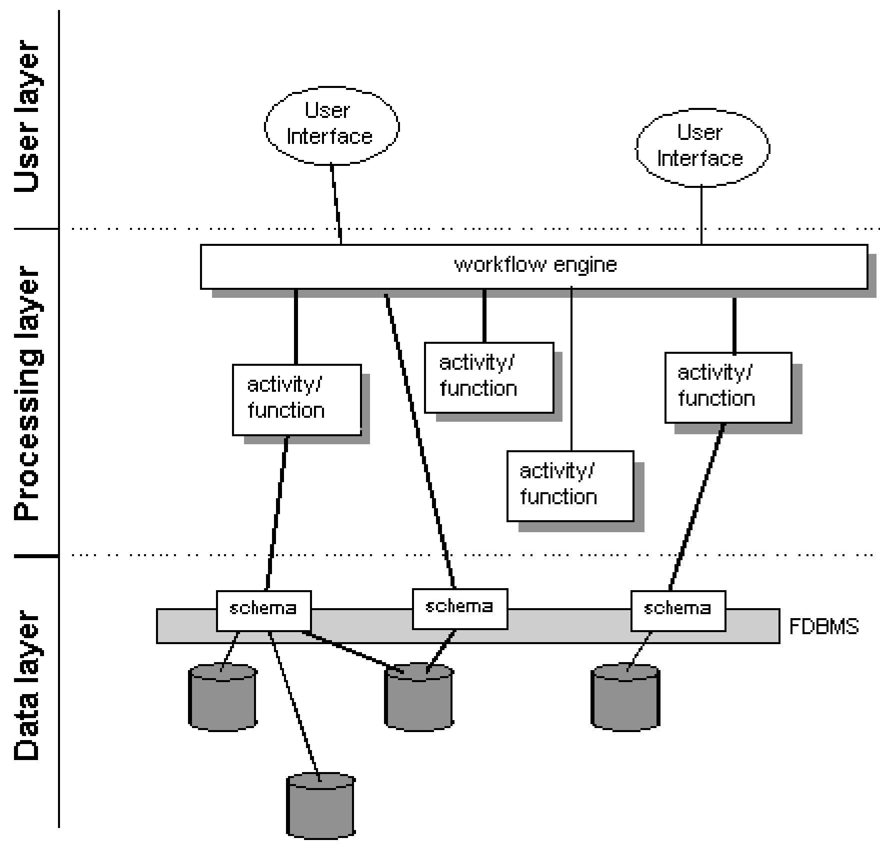


***Figure 2: FDBS and WFMS in OSCA***

This approach also puts a natural limit on the complexity of function building blocks. Since these building blocks access the data through the federated database system, their transactions are limited to those supported by the federated database system. More complex business process transactions are supported by combining function building blocks by means of the workflow manager.

## Change management

We briefly illustrate the flexibility of the proposed architecture with respect to change by examining the actions to be taken upon changes to a data building block, to a function building block, and to a business process

A data building block can be ported to a new implementation platform without changing the information content. Such a change will be completely invisible to the function building blocks, since this is completely dealt with by the federated database. In the case of merging data building blocks or splitting them, there will be no effect on the function building blocks. Changes that involve change of information content (e.g. schema evolutions) may affect function building blocks. To what extent depends on how much of the change can be covered in the federated database and the information need of the function block. Changing a function block will only affect those workflows that contain the specific function block. To what extent the workflow description needs to be changed depends on level of change. An option that eliminates the need of changing the workflows is by introducing a new modified function block while leaving the old one in place. Changing a business process will lead to a change of the workflow description for that process. Depending on the availability of the required data and functions, new data building blocks and function building blocks may be needed.

As one can see, the architecture always limits change to the building blocks involved. Changes are in terms of building blocks and not in terms of entire applications or entire information systems.

## Conclusions

We have presented an extension of the OSCA architecture that includes workflow management systems and federated databases. The resulting architecture provides uniform access to collections of data building blocks as well as to collections of function building blocks. We have outlined how mixtures of basic database transactions and advanced business process transactions can be supported and we have illustrated the flexibility of the architecture with respect to changes in data, functions and processes. We are well aware that quite a few issues have to be solved in order to realise the architecture described in this paper. Nevertheless, current commercially available technology allows medium scale experimentation with the approach.

## References

[1] R. Agrawal, D. Lomet; *Special Issue on Workflow and Extended Transaction Systems*; IEEE Data Engineering Bulletin, June 1993.

[2] Brodie, M. L., *The Promise of Distributed Computing and the Challenges of Legacy Systems.* In: Gray, P. M. D.; Lucas, R. J. (eds.): LNCS 618, Springer-Verlag, 1992.

[3] O.A. Bukhres, A.K. Elmagarmid; *Object-Oriented Multidatabase Systems*; Prentice Hall, 1996.

[4] U. Dayal, M. Hsu, R. Ladin; *A Transactional Model for Long-Running Activities*; Procs. 17th Int. Conf. on Very Large Databases; 1991.

[5] A.K. Elmagarmid (ed.); *Database Transaction Models for Advanced Applications*; Morgan Kaufmann; 1992.

[6] H. Garcia-Molina, K. Salem; *Sagas*; Procs. ACM SIGMOD Int. Conf. on Management of Data; 1987.

[7] P.W.P.J. Grefen, R.N. Remmerts de Vries; *A Reference Architecture for Workflow Management Systems*; Memorandum INF95-36; University of Twente, 1995.

[8] P. Grefen, J. Widom; *Protocols for Integrity Constraint Checking in Federated Databases*; Procs. 1st Int. Conf. on Cooperative Information Systems; 1996.

[9] F.J. van der Linden, J.K. Müller; *Creating Architectures with Building Blocks*; IEEE Software, November 1995.

[10] Bellcore, *The Bellcore OSCA Architecture*, Techn. Ref. TR-STS-000915, Issue 1; 1992.

[11] Sheth, A.; Larson, J.: *Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases*. ACM Computing Surveys, Vol. 22, No. 3, 1990.

[12] G. Wiederhold; *Mediators in the architecture of future information systems*; IEEE Computer, Vol. 25, No. 3, 1992.

[13] Glossary - A workflow Management Coalition Specification; Workflow Management Coalition; 1994.