Research Issues in Workflow Systems *

Jari Veijalainen, Aarno Lehtola, Olli Pihlajamaa
VTT Information Technology
P.O.Box 1203, FIN-02044 VTT, Finland.
Telephone: +358-0-4566014. Fax +358-0-4567028.
Email: {jari.veijalainen,aarno.lehtola,
olli.pihlajamaa}@vtt.fi

October 2, 1995

Abstract

Recent recession has compelled many companies to find more effective ways to conduct their business. One remedy suggested is to model organizational dynamics as "business processes" and provide a suitable tool support for this. In this context the buzzwords business process re-engineering and workflow have been often quoted. The basic idea in this thinking is to view the functioning of an organization to consist of business processes and provide computer support for as large part of the processes as possible through "workflow systems". There are currently perhaps hundreds of products which claim to support workflows. We believe that workflow techniques are a proper way of supporting the process approach, but that the approach requires further development for several reasons.

In this paper, we will discuss the development needs by presenting organizational and architectural considerations and requirements resulting from them for the workflow techniques. One central observation is that organizing a significant portion of work in organizations as business processes will sustain also in the future and thus the workflow support with all its ramifications will be needed. The future form of the support depends, however, largely on technological and organizational development. In a way, the key issue is how well the investments in current - and future - legacy systems can really be exploited by new generations of workflow systems or workflow components. We will pinpoint research issues as appropriate and present some novel ideas e.g. on why and how workflows can be made more modular and reusable in different contexts, and how transactional features could still be incorporated into workflow techniques in such a way that they better serve the process abstraction and simultaneously are an integrating technology on top of legacy systems. We also address the impact of new technologies,

*This work was done in the ESPRIT III LTR project TransCoop (EP8012) which is partially funded by the European Commission. The partners of TransCoop are GMD (Germany), University of Twente (The Netherlands, and VTT (Finland).

such as fast networks and extensive use of multimedia data on their development, as the current workflow systems might be legacy systems of the future.

1 Introduction

The development of organizations and technology are in a complicated relationship with each other, both being conditions or prerequisites for each other. Especially the recent development of information and communication technology has had a deep impact on the organizational structures and the work in the organizations. During the past 30 years, many tasks have disappeared or their content transformed (cf. typist) and completely new jobs (cf. microcomputer specialist) have emerged. Information processing has been to a large extent computerized in the industrialized part of the world and currently the telecommunication technology is becoming an integral part of the common information processing infrastructure of the organizations and the whole society. This is the overall situation we have in mind when discussing the workflow technology and its research trends below.

From the point of view of an individual organization the past is also important, when introducing new technology. The investments of organizations into data processing equipment and software are significant and there is a big interest to benefit as long as possible from the existing legacy systems, which often significantly resist modification and evolution. On the other hand, the recent recession has compelled many companies to find more effective ways to conduct their business. As a result of this, one has become aware of the fact that much of the organizational dynamics can be modeled as processes and the buzzwords business process re-engineering and workflow have been introduced.

What is a workflow system? The basic idea behind "workflow systems" is that an organization can be seen to perform (information handling) "processes". Each instance of a process of a certain type is by structure identical or quite similar to another instance. Typical examples of these kind of processes are travel bill handling or customer order handling within an organization. Because the instances are identical or similar, one can describe the process type for each of them.

The second, and a newer idea, behind a workflow system is that the process types are described in a form which can be "understood", i.e. interpreted by a computer. Computerization of the process description and execution makes several things possible, like controlling the process instances (temporal restrictions, selection of persons performing the tasks/steps) and tracking their state by computers, but also verification of the consistency of the process (cf. process definition tools below).

Computerization also makes possible the third important idea, namely that the information to be handled during the process can be combined into the computerized process instances. Thus, a travel bill handling process instance does not only consist of the description of which steps have been so-far done by whom and which are pending, but it also contains the travel bill form itself, and as much electronic annexes, (cf.a metro ticket) as possible. There is one persistent problem visible, namely that not all information handled by a process is computerized, but still it should be accessible during the handling process.

The fourth important idea behind a workflow system is that such a system can be used for *coordination* of the work in a decentralized organization. Seen from this perspective, a workflow system is a "glue" system in two sense. It glues together individuals working in a process at different locations but it also glues together the computer environments used in different parts of an organization - or in different organizations. A necessary prerequisite for this role is that the computers used by the workers in a process are connected by a network capable of carrying the necessary data and offering a suitable protocol support and security.

A workflow specification or definition is the representation of the type of the process, consisting of tasks (steps, activities) and their relationships, often called dependencies [ASSR93], along the data (types) handled by the process. The specification and its relationship with the system, application, and the application data are crucial when differentiating between workflow system generations. Historically, workflow systems have developed from at least two directions, office procedure systems and image management applications [AS94]. The first generation systems were applications of a particular application area (e.g. image or document management). The process specification was hardcoded into the application program(s). The second generation systems factored out the workflow capabilities and offered a script language to represent the workflow specifications. They also offer a limited selection of third party tools, like editors or databases, but the workflow system remains mainly an application among other applications. A typical example of this kind of a system is Staffware [Sta94a, Sta94b]. In third generation systems the workflow specification is given separately, usually through a graphical interface and the essential component is the workflow engine which interprets the specifications. The interfaces allow many other tools to be used and the engine can also be used through APIs making the system to a "glue". The interchange formats are proprietary, as well as the interfaces and process specification language. This is the current stand of the things, and there are tens if not hundreds of workflow products which have features of second and third generation, and more are coming.

The fourth generation is coming. One vision for the fourth generation systems is that they will become part of the middleware and will offer services among other services. Interfaces and interface formats will be standardized and there will be a standard process definition language.

In some sense what follows is an attempt to find out how could such a fourth generation workflow system look and what should be researched into in order to realize them. We approach the issue in the following way. First we analyze what properties real-world usage requires from workflow systems. In principle, this sort of analysis is multi-dimensional and it is not possible to locate each requirement into one place only. We have divided the analysis into four parts depending on factors affecting to the way an organization works. We consider requirements which are brought about by operational goals, organization structures, technical frameworks, and business environments of organizations.

After presenting the external requirements, we consider how they affect to the internal realization of the current workflow management systems. We structure this part of presentation according to the reference model of the Workflow Management Coalition

(WfMC) [Coa95] (see Section 3.2). Moreover, we assume the workflow-based approach to process modeling to consist basically of a specification step of a workflow using some process definition tools and the subsequent recurrent execution of the specification by some workflow enactment service. The definition phase is supposed to include process simulation and validation tools.

In section four we then look into the future, i.e. we grasp how the fourth generation and beyond workflow systems could look like. There we try to guess how technology might develop and what impact it might have on these systems and services. Our belief is that standard process specification language or languages will develop and toolsets will be developed to support them. We also discuss the mail-versus database driven approaches, as well as document versus process-oriented. Section 5 concludes the paper.

2 Organizational Requirements

In this section we outline requirements an organization may set to its workflow management system. Organization structures affect on the way work is internally carried out in organizations. Static line hierarchy and dynamic team organization can be regarded as the extreme cases. Same operational goals can be met with different organizational structures. Workflow systems emphasize the role of processes as central organisational entities. It is interesting to analyse how the continuous change within today's organizations affects to the work processes and poses new challenges to the workflow management systems in use. Some features of processes can be regarded as invariant during the organizational changes, while some of the features are new.

Dynamics of workflows: Most of the current commercial workflow products concentrate on relatively static workflows in quite rigid organizations. There are many cases where workflow automation could be profitably applied, but where the basic requirements surpass those covered by available systems. In some application domains, there is need for flexible process definitions. In the most extreme cases, the process definition is written along the execution of the process. For instance, in many design tasks, this sort of behaviour is frequent when teams of designers are working. Subtasks are dynamically launched whenever need is identified. In case of designing complex systems such as ships and paper mills, the overall process may even involve multiple organizations.

In most workflows, there can be exception situations when e.g. work has to be quickly redistributed. With regards to workflow management systems this means that system administrators must be able to gracefully intervene active process instances and conduct changes into them, such as reassign activities to new actors. Management operations should be executed without reducing the reliability of the overall execution.

Autonomy: Autonomy is an issue which directly or indirectly causes many of the problems faced when developing and using workflow systems. The reason is simple, they cross the borders of autonomous organizations or relatively autonomous organizational units. Intuitively, organizational (O-)autonomy means that the organization cannot be controlled by another through (some) interactions. The organization has thus self-determination in this respect.

O-autonomy is the sourc which are of interest here, like design (D-), management (M-), communication (C-), and execution (E-)autonomy. A D-autonomous organization is able to determine itself how its computer hardware and software architecture telecommunication infrastructure etc. is composed. This leads easily to heterogeneity between the technical infrastructures of different organizations and even organizational units. M-autonomy means that on organization can determine itself how the systems are used (policies, security, level of service, etc.). One of the main consequences of M-autonomy is that computers (e.g. servers, but especially clients) and other end systems (e.g. mobile phones) can be unreachable through a network. Thus, they behave in an C-autonomous manner. Another consequence of M-autonomy is E-autonomy, which means that a computer does not need to perform at all measures indicated in a request or can execute the measures at its own pace and in the way best suited for the organization.

As discussed in [Vei92] (see also [Vei93]) a solution to the problems incurred by heterogeneity and autonomy is to establish a *homogeneous* global domain. In the workflow area, this task is performed e.g. by WfMC.

Because autonomy is the essential attribute of an organization or organizational unit, we will consider autonomy and its different expressions and consequences as an invariant which influences the technical development of workflow systems.

Modularity and versioning of specifications: Generally, work processes can span over D-autonomous organizational units. How will the specifications be organized? A solution is to make them modular and let those organizational units design and specify the modules who are responsible for the tasks. The interoperability of the different modules must be preserved by suitable means. Modularity also promotes reuse in specifications.

In large organizations there can be thousands of different processes or their parts in execution as workflows. These probably undergo continuous development cycles. As the processes often are long-living, there can be several versions of the workflow specifications and instances present simultaneously. The workflow environment must support different versions both for specifications and their instances.

Systems integration: Work processes of an organization often integrate both manual and computer supported tasks. For instance, while processing incoming invoices the original documents usually are in paper form, and before entering to the computer supported part of the handling process, the contents of the paper documents must be input to the systems.

An important aspect of workflow systems is to regard them as "systems glue". Workflow management systems should interoperate with existing legacy systems, communications facilities, and other systems of the organization in order to flexibly support the overall working process of the organization. Migration of existing legacy systems is an important research topic [BS95].

Some workflows are very data communication intensive. For instance, the PortNet EDI system for Finnish ports [JLP$^+$95]. The workflow management system is expected to interface properly to a set of data communications facilities.

From architectural perspective, the framework can be perceived as follows: There are three layers in the runtime system, where the global homogeneous workflow layer effectively connects different component systems. The local component systems form the lowest layer. They can be legacy databases or other data repositories like file systems and legacy applications running on top of them. They can also be mobile units, like portable computers connected over a GSM-link to the home base or other mobile units. The interface between the global and local layer in a system converts the global homogeneous operations and data into the local format and vice versa. A further requirement arising from technology development is the need to combine multimedia data types into the workflow environments.

Locale and business specific needs: Business environment refers to the surrounding reality where an organization operates. The environment implies business practices, standards, legislation, official guidelines, cultural habits etc. to which the organization must adapt in order to do business successfully. These may vary much in different places of the world. Work processes of an organization must easily adapt to the changes in the business environment. This requires flexibility from the used workflow management systems.

Business practices determine rules used in business transactions, such as competitive bids. ISO 9000 series of quality standards are a good example of standards influencing to the work processes by requiring e.g. well documented and later on tractable execution. Legislation may pose requirements such as long term archiving of central documents and good tractability of the processes to support re-examinations. Security operations are important in business environments. These involve data encryption and user authentication and authorization checking services.

3 State of the Art

3.1 The evolution of workflow management products

Workflow management software originates from the early 80's when some of the imaging system vendors added routing and tracking functionality to their products. Although the data routed were first only digitized images it was soon realized that such routing capabilities could be a great help for managing any kind of data in corporate processes. Workflow management was pulled out from the imaging applications thus yielding a new branch of software. Another origin for workflow systems is the pioneer work done in office procedure systems at the same time [AS94]. However, the exploitation of the workflow technology started slowly because the office environments in those days were not extensively computerized.

The first wave of the office automation aimed to support and increase the efficiency of the work done by an individual. The second technology wave networked office workers' workstations founding the technological basis for the business process automation. Still, a push to new way of thinking were needed.

Finally, in the early 90's it was discovered that automating single business tasks does not increase the productivity considerably but automating of a whole business process would be needed instead. Workflow management became a key issue because it was expected to enable the management of any kind of business processes. Workflow

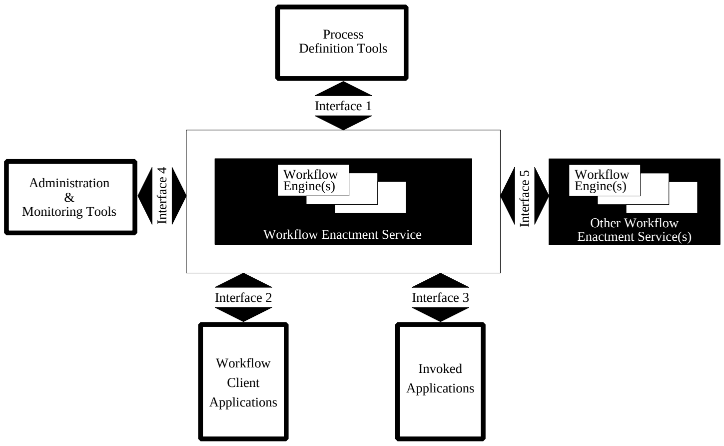

Figure 1: The WfMC's Workflow Reference Model

management was also seen as a solution to interoperability problems between "islands of automation" in larger corporations.

Currently, there are hundreds of vendors in the software market claiming that their products have workflow capabilities. Most of the corporations are now doing or intending to do business process re-engineering using varying strategies and methods in order to automate their business processes using varying strategies and methods. Still, the lacking standardization in the field of workflow management systems hinders the interoperation between different workflow systems as well as integration of workflow systems with other IT services. Such standardization work is in progress by the Workflow Management Coalition.

3.2 Standardization work by the Workflow Management Coalition

The Workflow Management Coalition (WfMC) was founded in August 1993 "to center gical promote the use of workflow through the establishment of standards for software terminology, interoperatibility and connectivity between workflow products" [Coa95].

The standardization work is centered around the *Workflow Reference Model* (see Figure 1) [Coa95, Hol94] which aims to "specify a framework for workflow systems, identifying their characteristics, functions and interfaces" [Coa95].

of the model is the *Workflow Enactment Service* which provides the runtime server capable to create, manage and execute workflow process instances. The Workflow Enactment Service may consist of one or more *Workflow Engines* providing the runtime environment. In case of multiple engines process executions are partitioned to different engines in the enactment service following some partitioning logic (e.g. based on the workflow type).

The Workflow Enactment Service is surrounded by Workflow Application Programming Interface (WAPI) divided to five different interface areas (interfaces 1-5 in the

Figure 1. The sets of functions belonging to these different interfaces are partially overlapping. Hence, the WfMC is now tending to consider the WAPI as a one unified service interface that provides functionality to the following kinds of workflow components:

Process Definition Tools. These tools are used for analyze, model, and describe business processes. Interface 1 provides common interface format for importing/exporting the business process definitions produced by the Process Definition Tools. This makes it possible to transfer process definitions between different products, allow to define and execute processes using separate tools and bridge the gap between descriptive models produced by the BPR-tools and executable workflow process definitions run by the workflow management systems.

Workflow Client Applications, using Interface 2, provides the end-user interface to the Workflow Enactment Service (usually called a *Worklist Handler*) allowing users to select the tasks, retrieving details about the work to be done and invoke applications needed to perform the work among the other things.

Invoked Applications consists a wide range of pre-existing services in the heterogeneous environments that must be integrated to the workflows. Interface 3 enables also the invocation of automatic activities and *workflow enabled* applications which are specifically designed to interoperate with the Workflow Enactment Service.

Other Workflow Enactment Services. One of the primary concepts in the work of the WfMC is to provide interoperatibility between heterogeneous workflow enactment services allowing parts of the process execution to be passed to other workflow enactment services. This is done through the Interface 4.

Administration and Monitoring Tools. Interface 5 allows to administer all the Workflow Enactment Services with the same tool. In the same way, monitoring and analyzing all the workflows (possibly managed by different workflow products) would be enabled with one and the same product.

The WfMC concentrates on the interoperability and communication standards. Hence, the functionality of the workflow products and the possible workflow models the products are based on remains still largely decided by the workflow product vendors.

In the following sections (3.3 - 3.6) we present some issues to be considered in the workflow research and development of the current workflow products following the structure of the Workflow Reference Model introduced above. We also point out some ongoing and already done research work.

3.3 Research issues in process definition tools

A graphical specification of workflows is possible with many commercial workflow products, but hardly any of them provides analysis and verification tools for the specifications. On the other hand, there are dozens of separate Business Process Re-engineering tools which provide such features. The problem is that the descriptive models produced by current BPR-tools cannot be automatically transformed to the executable process definitions used by the workflow systems. We see the linkage of the BPR-tools and the workflow systems very important. Here the common interchange format (Interface 1 in Figure 1) to be developed by the WfMC is in the key position.

In our research so far, we have found out that process definition tools, whether they are part of the workflow product or form a separate BPR-toolset capable to interchange definitions with workflow products, should support:

- verification of transactional correctness of process definitions, and
- performance analysis of a set of workflows in terms of throughput/response time,
- estimation of resource requirements (system, people) to run the workflows with certain throughput/response time to satisfy customer needs.

These are important requirements from practical point of view, as they make possible dimensioning of the resources. It also makes evident that reasonable performance measures must be defined, based on which the processes can be evaluated and improved. Especially transactional correctness of the process definitions is a new research issue.

Another practical research issue is find out how to maximize re-usability of the already existing process definition building blocks. Although hierarchical modeling and the use of object oriented technology inherently support re-usability we feel that there are still work to be done in this field. For example, how to provide different transactional properties to the same activity instantiated in different process contexts.

3.4 Research issues in workflow enactment services

We perceive this module primarily as a run-time environment for workflows. Therefore, almost all aspects relevant for workflows have implications here.

3.4.1 Transaction model support

Transaction models were first developed for database systems and a basic theory emerged [GLPT76, EGLT76, HR83, GR93]. The correctness of interleaved program executions was defined by (RW-)serializability [BHG87, Pap86].

There has been a lot of research where the basic RW-model has been extented or replaced by a more general model where, e.g., methods or operations are addressed and more general models of computation than RW-sequences are used. Many of them are interesting as a starting point for considering transactional properties of workflows.

It is evident that workflows also benefit from transactional properties, especially such is atomicity which guarantees that an execution ends in a successful or an unsuccessful end state, but does not stop in an intermediate state. Concurrency control which has been the main focus of the research can be partially taken care of at the local database level. At the enactment service level the global concurrency control should be taken care of. The research issues here are what kind of concurrency control is needed, what is achieved in terms of data consistency, and with which kind of mechanisms it is achieved.

Nested transactions [Mos82, Lyn83, Mos85] were among the first extensions. There, the computation has a tree structure ending at RW-leaves. The same holds for multi-level transactions [Wei86, WHBM90, Wei91], open Nested transactions [WS91, MRKN92, WS92] and S-transactions [Vei90, VEH92]. In open nested transactions the operational semantics is drawn into consideration as a criterium for commutativity at the higher levels of the execution trees. In S-transactions the main focus is on autonomy and the recovery is based on compensation [GM83, KLS90, Vei90], as is the case also with open nested transactions. Sagas [GMS87, GMDGKS90, GMDGKS91] are simple S-transactions and the Flex Transaction Model [BEK93] also bears some similarities.

Split-transactions [PH88, KP92] represent a novel view of how to dynamically restructure transactions. This idea might be of interest also for workflows. Transaction Specification and Management Environment (TSME) [GHM+93, GHKM94, GH94] is a complete approach directed towards workflows.

As an implementation technique for transactional and other features the ECA model is of interest [DBM88, DHL90, DHL91] and as a specification environment ACTA [CR90, CR91, CR92, CR94].

There have been some projects, prototypes and products, where transactional features are addressed, notably the work related to the (further) development of IBM FlowMark [LA94, LR94] done in Exotica project [MAGK95, MAA+95, AGK+95, AAA+95] and by Leymann [Ley94]. Of prototypes METEOR [KS94, KS95] and WAMO [EL94, EGN94, EL95] contain significant contributions in these directions, as well as Carnot [WPAM+93], InConcert [MS93], ASSET [BDG+94] and TriGS$_{flow}$ [KLRSR95].

3.4.2 Implementation techniques for long-living and real-time scenarios

Workflows often run long periods of time, weeks or even months. Combined with the other requirements above, like quite recurrent updates of the specifications, and transactional properties this necessitates a careful design of the run-time system. Clearly, certain degree of persistence for the computations must be guaranteed, in order to be able to recover forward and backward after crashes. A similar thing holds for specifications; they should exist as long as there are workflow instances running based on them. This requires a versioning mechanism to be set up which reflects the number of instances linked to it.

In administrative environments an acceptable response time is not difficult to achieve, provided the load is reasonable. There might still be problems, if the workflow system is the central component and heavily used. System performance meters and metrics are thus a relevant issue.

It seems that workflow techniques could be used also in environments, where the computations have soft or even hard real-time requirements (cf. telecommunication networks). In this case, implementation techniques must evidently look different from the above. E.g. the atomicity might not be a problem from application point of view, but rather throughput. Then, persistence is not necessarily a desired property.

It seems that real-time and non-real time workflows have quite different implementation requirements. These and their relationships are a research issue.

3.5 Research issues in workflow clients and invoked applications

One of the most important practical issues in the workflow management is the integration of the external world to the workflow system. Workflow systems are now seen as umbrella applications integrating everyday office software, corporate databases and other services in a controlled way. An important subproblem is the integration of legacy systems.

Hiding heterogeneity is only one question and not even the most difficult one. Autonomy issues especially in inter-organizational environments make it difficult to specify and enforce rules and properties needed to ensure correct process executions. For instance, preserving certain transactional properties of the process instance may be impossible if autonomous databases are involved.

When interfacing workflow enactment services with external applications, there are many ways the interface protocols may look like. How tight is the coupling of the programs? What is the granularity of the service requests? Is the interface protocol a generic one or purposely designed for a particular workflow application? Does the protocol support mixed initiative requests or is it asymmetric with this respect? The last question refers to the consideration, are we dealing with client applications and server applications, or rather agent applications with communication skills of the both.

In the following we consider a pragmatic example on interfacing. First we consider tightly coupled interfacing in a workflow to MS Excel for Windows spreadsheet program (shortly Excel). Technically, the protocol could be based either on Object Linking and Embedding (OLE) protocol or on Dynamic Data Exchange (DDE) protocol, which are both proprietary to the Windows environment. An application program can have both a server and a client role at the same time. In the case of using Excel on a fine level of granularity, the operations required by the workflow (e.g. filling in some particular cell) could be predefined using, e.g., Excel macros, which could then be invoked through a DDE link. The operations could be made to address a spreadsheet on cell level and perform very detailed things. A reverse DDE link could be used to note about an event taking place in Excel, e.g., the workflow system could be notified of changes of spreadsheet cell values via messages. In this case the protocol would be very scenario-specific and full in details.

In loosely coupled interfacing, Excel could be invoked just to edit one whole spreadsheet. After finishing the editing the user is responsible for saving the document. Most products provide this type of interfacing. If the external program is a legacy system, the protocol could be based on, e.g., pseudo terminal or access to its database. The transfer of bulk data could be also realized via a shared file system.

3.6 Research issues in administration and monitoring tools

Automating business processes is a continuous process and it usually doesn't stop to the implementation/generation of the executable process definition. Even though we had good simulation and verification tools in the definition phase it is not possible to detect all the potential bottlenecks coming up in the production use. For this we need comprehensive performance monitoring with customizable metrics and possibility to use the monitored data as input in the process definition phase or to automatically alter the execution of the processes (e.g. work balancing). In the current systems the loop back from the monitoring to the definition or execution phase rarely even partially exists. Hence, methods and implementations for coupling monitoring phase with definition and executing phase should be further investigated.

Most of the data needed to build an organizational model for the workflow system is already stored in the pre-existing (legacy) databases (personnel management systems). Much of the redudant work would be saved if these data storages could be exploited instead of trying to maintain the same data in many parallel systems. Similarly to integration of workflow activities and legacy systems we have open question here: how far we can go in integration?

4 Long-term research issues

By long term research issues we refer here to issues which target the problems of the workflow systems in use after the year 2000. It is of course quite difficult to guess which trends will prevail in 2000 and beyond as the pace of the technological development is very fast. The first question is whether workflows are an issue any more. They might become an uninteresting issue mainly for two reasons; either they might become as common a service as electronic mail today, or they might become obsolete. The latter alternative is hardly to be expected, as the process approach seems to bring benefits for organizations and the competition will force all organizations to follow it.

4.1 The future of the WfMC reference architecture

The hard core of the workflow based approaches seems to be the process concept. We believe that is description survives as a separate entity, the reasons being that organizing the work into processes is independent from any particular computer application, and that the process approach seems to be an invariant. Organizing the enactment service as the central component in the architecture renders and making the distinctions between the different funtionalities makes it plausible that the architecture will survive.

To describe the processes in a precise manner, a workflow specification language understood by a wide audience is needed. Currently, there is no commonly accepted language to describe the processes, although different products offer different ways of representing them and WfMC also has applied a graphical notation [Wor94].

It is a research issue whether there is only one specification language and what it is like, or is it still impossible to cope with only one language in different organizational and computer environments with different needs.

Because the processes are easy to describe with graphical symbols there will evidently be a graphical form and correspondingly graphical user interfaces for the description language(s).

4.2 Enactment service as part of the middleware

The second permanent component following from the invariant above and the separation of flow control from data is the enactment service which is able to interpret or “run” the specifications given by a specification language.

There is a set of architectural issues, which we consider to be important research topics:

The current generation of workflow engines can be considered as a separate software package. In the future, it is anticipated to be incorporated into middleware. One scenario is that it remains as a separate piece of software, but it is able to interwork with the other middleware components, like email and databases. It might become an add-on to an electronic mail component, but such an enactment service has its limitations. It is naturally a document driven architecture and using complicated process structures with loops become a problem. Also providing persistence of the data is not given as there is usually no database support. Email-based solution is, however, inherently distributed.

The other direction where the enactment service could find its place is the database system software. In this case, the database management system would offer enactment interface and the other four interfaces of the reference architecture at its interfaces. It is a research issue, whether and how the interfaces of the reference architecture could be realized as part of the database management software. Another similar approach is to implement the enactment service as a special application on top of a database management system. In this case the interfaces are separate from those of the database management system.

Another research issue is how the process definition tools look like in different settings of the enactment service.

The object technologies are currently gaining momentum in software engineering. It is our belief that their development play an essential role also here. The WfMC has already taken up the trend.

4.3 Symmetry/asymmetry and layering in the architecture

The reference architecture of the WfMC contains the idea that there are workflow client applications and invoked applications. This is natural, because the process description is a separate entity and the enactment service interpreting it is correspondingly an independent component. Different enactment service components can be in in different relationships with each other, i.e. they might be in a client-server relationship or in a symmetric relationship, like agents. The same applies to the relationships between the client/invoked applications and the enactment service. It is research issue, what is the best way to do it under different assumptions.

Replicability and/or scalability of the architecture are important issues. It is evident that one enactment service component cannot serve 300000 users, but some kind of replication is needed. The large scale workflow systems casts up many research problems (like robustness, performance, replicability of data and processes, crash recovery, etc) which should be addressed. They have one criteria in common: good solutions are scalable.

Layering of the workflow system is an important way of managing the scale. One important issue is what is the role of client/invoked applications and the system, i.e. application architecture. Especially the mobile computing environments bring up this question; should one of the applications run in the laptop? Or should the enactment service run there, too? In the latter case implementation techniques for the enactment service become important.

In all cases above, the protocol stacks supporting workflow systems are of importance. Although for text document shipment current protocols will do, the new data types (video, audio) might require new solutions (see below).

4.4 Heterogeneity and autonomy of component systems

This is also an invariant of the environment in question. When big workflow systems are set up, autonomy and heterogeneity of the components become visible evident. The interoperation of workflow systems over the organizational borders is important on one hand, but difficult on the other. As discussed above in section 2, homogeneous global area is needed. The question here is, what are the targets to be homogenized. Currently it seems that the standardized specification language is of prime importance here. It should be possible to port it from one enactment service to another or at least guarantee interoperability between them. It might be also necessary to ship specifications from one enactment service to another and execute them on the fly.

The mobile computing environments bring the need to master the C-autonomy [Vei90, Vei92] within the architecture. Laptops should be integratable with the other components.

Autonomy always brings up the security aspects of the systems. Who has right to do what? How is the policy enforced? If such mobile units as laptops are connected to the workflow environment, security is certainly an issue, but no matter which network is used to let external access, security remains an issue.

4.5 Legacy systems of the future

What are the legacy systems in 2000 and beyond? One such system might be the world-wide-web (WWW) and the data in it. The amount of data in this currently file-based system is rapidly increasing and there is no clear vision, how this data could be used also in workflow. A permanent system class which will persist is databases. In the years to come, object-oriented ones might gain momentum. Therefore, the design of the workflow systems should be such that data especially in object-oriented databases and should be accessible from the workflow systems.

From architectural point of view, the client/invoked applications should have access to or be part of legacy systems. Only so the investments can be saved.

4.6 Future networks and multimedia data

Currently, networking technology is taking big steps in increasing the transmission speed. It makes it realistic to transfer big chunks of data, e.g. audio and video stream, from the other side of the globe. On the other hand, the telecommunication networks and computer networks are not any more so far away from each other than they used. The digital convergence will evidently win.

Workflow technology could be used as an enabling technology of the overall communication infrastructure, especially in intelligent networks. There, the requirements are quite different from those found in business applications. People are not directly involved, but the workflow is fully automated. The structure of such a special purpose workflow is presumably simple, but the enactment service has to guarantee real-time properties and high throughput. Transactional properties might be of importance here (consider e.g. recording the location of a mobile phone; it must be consistent in different parts of the network, which stresses atomicity of executions).

Whereas information infrastructure represents a novel application field, multimedia data represents new data type in a rather old application field. It is quite easy to imagine that they come together. Consider such a simple thing than audio annotations. A customer phone call can be recorded in a digital format and annexed as data to workflow. In a similar manner still images or even video clipses could be handled as data document during the workflow.

It is largely unexplored, which consequences handling large multimedia documents and presenting them with hard real time requirements architecture of workflow systems and their components. For instance, the specification language might need reworking, as well as the overall architecture.

5 Conclusion

We have looked at research issues in workflow systems. A central plausible assumption seems to be that organizations are moving towards process-centered approaches when organizing their work. The workflow systems as an implementation support for these will, therefore, exist also in the future. Second assumption is that processes are and will be handled by relatively autonomous organizational units. This makes autonomy and its ramifications another invariant issue in workflow technology. We try to pinpoint, what kind of research should be done in order to arrive at next generation workflow systems. We start by examining the current state of workflow approach in organizations and some requirements. Most of the available workflow managing products are targeted to support well-defined administrative routine processes in big, structurally relatively stabile organizations. They do not fulfil all the requirements found. For instance, transactional properties are mostly lacking. This is especially true for email-based systems. The research here is progressing, but more is needed.

We then handle the issues raised by the state-of-the art reference architecture of the Workflow Management Coalition. What seems to be lacking is the orientation towards a complete and coherent set of tools which covers process modeling and process dimensioning support, as well as verification correctness and runtime support. It is still unclear, could there be just one workflow toolset to fulfil all the requirements mentioned, and just one workflow specification language that would suit both to workflow analysis and validation and to compilation for runtime system. More research is needed there.

We also discuss many architecture-related problems like the functionality of the different application types, etc. A major architectural concern is how legacy systems can be made to interwork with the workflow management systems. Only doing it, past investments' yield can be increased.

After the current problems we look into the future and by guessing some technology trends raise issues which should be investigated. The impact of fast networking and mobile computing, but also incorporation of multimedia data into the workflows seems the most probable and challenging problems which deserve attention of the research community.

Bibliography

[AAA+95] G. Alonso, D. Agrawal, A. El Abbadi, R. Günthör C. Mohan, and Mohan U. Kamath. Exotica/FMQM: A persistent message-based architecture for distributed workflow management. In *Proc. of IFIP WG 8.1 Workgroup Conference on Information Systems Development for Decentralized Organizations (ISDO95)*, Trondheim, Norway, Aug 1995.

[AGK+95] G. Alonso, R. Gunthor, Mohan U. Kamath, D. Agrawaland, A. El Abbadi, and C. Mohan. Exotica/FMDC: Handling disconnected clients in a workflow management system. In *Proceedings of the Third International Conference on Cooperative Information Systems (CoopIS-95), May 9-12, 1995, Vienna, Austria*, 1995.

[AS94] Kenneth R. Abbot and Sunil K. Sarin. Experiences with workflow management: Issues for the next generation. In *Proceedings ACM 1994 Conference on Computer Supported Cooperative Work (CSCW '94)*, pages 113–120, Chapel Hill, North Carolina, USA, Oct 1994.

[ASSR93] Paul C. Attie, Munindar P. Singh, Amit Sheth, and Marek Rusinkiewicz. Specifying and enforcing intertask dependencies. In *Proceedings of the 19th International Conference on VLDB, Dublin, Ireland*, 1993.

[BDG+94] A. Biliris, S. Dar, N. Gehani, H. V. Jagadish, and K. Ramaritham. ASSET: A system for supporting extended transactions. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, Minneapolis, Minnesota, May 1994.

[BEK93] O. Bukhres, A. Elmagarmid, and E. Kuhn. Implementation of the flex transaction model. *IEEE Data Engineering Bulletin*, 16(2), June 1993.

[BHG87] Philip Bernstein, Vassos Hadzilacos, and Nathan Goodman. *Concurrency Control and Recovery in Database Systems.* Addison-Wesley Publishing Company, 1987.

[BS95] M. Brodie and M. Stonebraker. *Migrating Legacy Systems - Gateways, Interfaces & the Incremental Approach.* Morgan Kaufmann Publishers, Inc., 1995.

[Coa95] Workflow Management Coalition. Workflow management coalition overview. http://www.aiai.ed.ac.uk/WfMC/overview.html, Sep 1995.

[CR90] Panos K. Chrysanthis and Krithi Ramamrithan. ACTA: A framework for specifying and reasoning about transaction structure and behavior. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, 1990.

[CR91] Panos K. Chrysanthis and Krithi Ramamrithan. A formalism for extended transaction model. In *Proceedings of the 17th International Conference on VLDB, Barcelona*, Sep 1991.

[CR92] Panos K. Chrysanthis and Krithi Ramamrithan. ACTA: The Saga continues. In Ahmed K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 10. Morgan Kaufmann Publishers, Inc., 1992.

[CR94] Panos K. Chrysanthis and Krithi Ramamrithan. Synthesis of extended transaction models using ACTA. *ACM Transactions on Database Systems*, 19(3):451–491, Sep 1994.

[DBM88] Umeshwar Dayal, Alejandro P. Buchmann, and Dennis McCarthy. Rules are objects too: A knowledge model for an active, object-oriented database system. In *Proceedings 2nd International Workshop on Object-Oriented Database Systems*, pages 129–143, West Germany, Sep 1988.

[DHL90] Umeshwar Dayal, Meichum Hsu, and Rivka Ladin. Organizing long-running activities with triggers and transactions. In *Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, May 23-25, 1990, Atlantic City, NJ*, 1990.

[DHL91] Umeshwar Dayal, Meichum Hsu, and Rivka Ladin. A transactional model for long-running activities. In *Proceedings of the 17th International Conference on VLDB, Barcelona, Spain*, Sep 1991.

[EGLT76] K. Eswaran, Jim Gray, R. Lorie, and I. Traiger. The notions of consistency and predicate locks in a database systems. *Communications of the ACM*, 19(11):624–633, Nov 1976.

[EGN94] Johann Eder, Herbert Groiss, and Harald Nekvasil. A workflow system based on active databases. In *Proc. Connectivity-94: Workflow Management - Challenges, Paradigms and Products*, Wien, München, 1994. Oldenbourg Verlag.

[EL94] Johann Eder and Walter Liebhart. A transaction-oriented workflow activity model. In *Proc. of The Ninth Int. Symposium on Computer and Information Systems, ISCIS IX, Antalya, Turkey, Nov. 1994*, nov 1994.

[EL95] Johann Eder and Walter Liebhart. The workflow activity model wamo. In Steven Laufmann, Stefano Spaccapietra, and Toshio Yokoi, editors, *Proceedings of the third International Conference on Cooperative Information Systems (CoopIS-95), May 9-12, 1995, Vienna, Austria*, pages 87–98, 1995.

[GH94] Dimitrious Georgakopoulos and Mark F. Hornick. A framework for enforceable specification of extended transactions models and transactional workflows. *International Journal of Intelligent and Cooperative Information Systems*, Sep 1994.

[GHKM94] Dimitrious Georgakopoulos, Mark F. Hornick, Piotr Krychniak, and Frank Manola. Specification and management of extended transactions in a programmable transaction environment. In *The 10th International Conference on Data Engineering, Houston, Texas*, pages 462–473, Feb 1994.

[GHM+93] Dimitrious Georgakopoulos, Mark F. Hornick, Frank Manola, Michael L. Brodie, Sandra Heiler, Farshad Nayeri, and Benjamin Hurwitz. An extended transaction environment for workflows in distributed object computing. *Bulletin of the Technical Committee on Data Engineering*, 16(2):24–27, Jun 1993.

[GLPT76] Jim Gray, R. Lorie, G. Putzolu, and I. Traiger. Granularity of locks and degrees of consistency in a shared data base. In *IFIP Working Conference on Modelling of Data Base Management Systems, Freudenstadt, Germany*, Dec 1976.

[GM83] Hector Garcia-Molina. Using semanntic knowledge for transaction processing in a distributed database. *ACM Transactions on Database Systems*, 8(2):186–213, June 1983.

[GMDGKS90] Hector Garcia-Molina, Johannes Klein Dieter Gawlick, Karl Kleissner, and Kenneth Salem. Coordinating multi-transaction activities. Technical report, University of Maryland, 1990.

[GMDGKS91] Hector Garcia-Molina, Johannes Klein Dieter Gawlick, Karl Kleissner, and Kenneth Salem. Modeling long-running activities as nested sagas. *IEEE Data Engineering Bulletin*, 14(1), March 1991.

[GMS87] Hector Garcia-Molina and Kenneth Salem. Sagas. In *Proceedings 1987 ACM SIGMOD International Conference on Management of Data*, pages 249–259, may 1987.

[GR93] Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques.* Morgan Kaufmann Publishers, Inc., San Mateo, California, 1993.

[Hol94] David Hollingsworth. Workflow reference model. Technical Report TC00-1003 (Draft 1.1), Workflow Management Coalition, Nov 1994.

[HR83] Theo Härder and Andreas Reuter. Principles of transaction-oriented database recovery. *ACM Computing Surveys*, 15(4):287–317, Dec 1983.

[JLP+95] J. Juopperi, A. Lehtola, O. Pihlajamaa, A. Sladek, and J. Veijalainen. Usability of some workflow products in an inter-organizational setting. In *In Proceedings of IFIP WG8.1 Working Conference on Information Systems Development for Decentralized Organizations - ISDO'95*, page 16 pages (in print). IFIP, 1995.

[KLRSR95] G. Kappel, P. Lang, S. Raush-Schott, and W. Retschitzegger. Workflow management based on objects, rules, and roles. *IEEE Data Engineering Bulletin*, 18(1):11–18, March 1995.

[KLS90] Henry F. Korth, Eliezer Levy, and Abraham Silberschatz. A formal approach to recovery by compensating transactions. In *Proceedings of the 16th International International Conference on VLDB, Brisbane, Australia*, pages 95–106, 1990.

[KP92] G. E. Kaiser and C. Pu. Dynamic restructuring of transactions. In Ahmed K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 8. Morgan Kaufmann Publishers, 1992.

[KS94] Narayan Krishnakumar and Amit Sheth. Specification of workflows with heterogeneous tasks in METEOR. Bellcore Technical Memorandum TM-24198, Bell Communications Review, Inc., May 1994.

[KS95] Narayanan Krishnakumar and Amit Sheth. Managing heterogeneous multi-system tasks to support enterprise-wide operations. *Distributed and Parallel Databases*, 2(3):155–186, 1995.

[LA94] Frank Leymann and Wolfgang Altenhuber. Managing business processes as an information resource. *IBM Systems Journal*, 33(2):326–348, 1994.

[Ley94] Frank Leymann. Supporting business transactions via partial recovery in workflow management systems. submitted for external publication, Oct 1994.

[LR94] Frank Leymann and Dieter Roller. Business process management with flowmark. In *Spring COMPCON 94*, pages 230–234, San Francisco, California, USA, 1994.

[Lyn83] Nancy Lynch. Multilevel atomicity – a new correctness criterion for databse concurrency control. *ACM Transactions on Database Systems*, 8(4):484–502, Dec 1983.

[MAA+95] C. Mohan, D. Agrawal, G. Alonso, A. El Abbadi, R. Günthör, and Mohan U. Kamath. Exotica: A project on advanced transaction management and workflow systems. *SIGOIS Bulletin (Special Issue on Business Process ManagementSystems: Concepts, Methods and Technology)*, 16(1), Aug 1995.

[MAGK95] C. Mohan, G. Alonso, Günthör, and M. Kamath. Exotica: A research perspective on workflow management systems. *IEEE Data Engineering Bulletin*, 18(1), 19-26 1995.

[Mos82] J. Moss. Nested transactions and reliable distributed computing. In *Proceedings of the IEEE Symposium on Reliability in Distributed Software and Database Systems*, 1982.

[Mos85] J. Eliot B. Moss. *Nested Transactions: An Approach to Reliable Distributed Computing.* The MIT Press, 1985.

[MRKN92] Peter Muth, Thomas C. Rakow, Wolfgang Klas, and Erich J. Neuhold. A transaction model for an open publication environment. In Ahmed K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 6. Morgan Kaufmann Publishers, Inc., 1992.

[MS93] Dennis R. McCarthy and Sunil K. Sarin. Workflow and transactions in inconcert. *IEEE Data Engineering Bulletin*, 16(2):53–56, Jun 1993.

[Pap86] C. Papadimitriou. *The Theory of Database Concurrency Control.* Computer Science Press, 1986.

[PH88] C. Pu and N. Hutchinson. Split-transactions for open ended activities. In *Proceedings of the 14th International Conference on VLDB*, pages 26–37, 1988.

[Sta94a] Staffware plc. *Staffware Technical Handbook for Version 5*, Apr 1994.

[Sta94b] Staffware plc. *Staffware User Manual*, 11 edition, Apr 1994.

[VEH92] Jari Veijalainen, Frank Eliassen, and B. Holtkamp. The s-transaction model. In Ahmed K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 12, pages 467–513. Morgan Kaufmann Publishers, Inc., 1992.

[Vei90] Jari Veijalainen. *Transaction Concepts in Autonomous Database Environments.* R. Oldenbourg Verlag, 1990.

[Vei92] J. Veijalainen. Issues in Open EDI. In P.A. Ng, C.V. Ramamoorthy, L.C. Seifert, and R.T. Yeh, editors, *Proceedings of the Systems Integration '92*, pages 401–412. IEEE Computer Society Press, 1992.

[Vei93] J. Veijalainen. Heterogeneous multilevel transaction management with multiple subtransactions. In *Fourth Intl. Conference on Database and Expert Systems Applications, DEXA '93, LNCS Nr. 720*, pages 181–188. Springer-Verlag, Sept. 1993.

[Wei86] Gerhard Weikum. A theoretical foundation of multi-level concurrency control. In *Proceedings of the fifth ACM SIGACT-SIGMOD Symposium on Principles of database Systems*, 1986.

[Wei91] Gerhard Weikum. Principles and realization strategies of multilevel transaction management. *ACM Transactions on Database Systems*, 16(1):132–180, Mar 1991.

[WHBM90] Gerhard Weikum, Christof Hasse, Peter Brössler, and Peter Muth. Multi-level recovery. In *Proceedings ACM-PODS, Nashville*, 1990.

[Wor94] Workflow Management Coalition. *Glossary; A Workflow Management Coalition Specification*, nov. 1994.

[WPAM+93] Darrel Woelk, Phil Cannata Paul Attie, Greg Meredith, Amit Seth, Munindar Singh, and Christine Tomlinson. Task scheduling using intertask dependencies in carnot. In *Proceedings of ACM SIGMOD Conference*, May 1993.

[WS91] Gerhard Weikum and Hans-Jörg Scheck. Multi-level transactions and open nested transactions. *IEEE Data Engineering Bulletin*, 14(1), 1991.

[WS92] Gerhard Weikum and Hans-Jörg Scheck. Concepts and applications of multilevel transactions and open nested transactions. In Ahmed K. Elmagarmid, editor, *Database Transaction Models for Advanced Applications*, chapter 13. Morgan Kaufmann Publishers, Inc., 1992.