Appeared in: *Proceedings of IEEE Intl Conf on Communication Technology*, Beijing:, Oct, 1998. S33-05-01 ~ S33-05-06

# Workflow Management Systems: A Survey

**SHI Meilin, YANG Guangxin, XIANG Yong, WU Shangguang**
{shi, ygxin, xyong, wsg}@csnet4.cs.tsinghua.edu.cn
http://cscw.cs.tsinghua.edu.cn
Department of Computer Science, Tsinghua University, Beijing, P.R.China 100084

**ABSTRACT** Workflow management systems can help business goals to be achieved with high efficiency by means of sequencing work activities and invoking appropriate human and/or information resources associated with these activities. As computer networks are used more and more widely in daily work, workflow management attracts more and more attention from both industry and research communities. This paper summarizes the state of art in WfMS research, detailing some workflow-related concepts and its typologies, the WfMS reference model proposed by WfMC, and some of the current research trends and hotspots.

**KEYWORDS** Workflow Management System Computer Supported Cooperative Work

## Introduction

As computers are used more and more widely in daily lives, people are more likely to rely on computers to get their work done. At the same time, many information systems have been built up to support or manage these work activities. However, running independently and with little awareness of the existence of each other, these separated systems lack the power to coordinate activities among multiple people so that their common business goals can be achieved with high efficiency. That is why workflow management systems (WfMSs) have attracted so many attentions from both industry and research communities as it enters 1990's.

WfMSs aim to coordinate activities and automate business processes by means of sequencing work activities and invoking appropriate human and/or information resources associated with these activities. Some of its basic ideas originate from *office automation*, d*ocument management, imaging processing, form processing* and other areas. From the CSCW [34] point of view, WfMS is a groupware system that supports asynchronous cooperation and can be viewed as the further development of groupware systems from supporting synchronous, unstructured cooperation to formal, structured cooperation.

## Some Concepts and Typologies

In order to understand what *WfMS* is, first we should know what *workflow* is. According to Workflow Management Coalition (WfMC[17], an international organization responsible for the standardization of WfMSs), workflow is concerned with *the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal*[41]. WfMS is thus *a system that completely defines, manages and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic*[41].

### WfMS Related Concepts

The following concepts will help us to get a better understanding of workflow and WfMSs.

- *Workflow model* (also referred as *process definition*) is the computerized representation of business process. It defines the *starting and ending conditions* of the process, the *activities* in the process, *control flow* (or *navigating rules*) and *data flow* among these activities, etc. The models implemented in different WfMSs are quite different. Early systems with workflow capability implement the workflow logic at the source code level, which makes it very difficult to modify the logic. Many other models, e.g., directed graph[24], conditional directed graph[39], PetriNet[4], Object-Oriented model[3,29], Language/Acts theory[31], Generalized Process Structure Grammar[28] (to name just a few) have been developed later. A good model should have sufficient description capability, be easy to use and be easy to be modified to accommodate to the varying work conditions.
- An a*ctivity* is a logic step within a workflow, which includes the information about the *starting and ending conditions*, the *users* who can participate, the *tools* and/or *data* needed to complete this activity, and the *constraints* on how the activity should be completed (such as the time limits), etc.
- A *process instance* is the execution of a process definition conducted by the WfMS. WfMS interprets the process definition and controls the instantiation of processes and activities and the sequencing of activities, adding *work items* to the users' *work lists* and invoking application tools as necessary. Data manipulated by application tools are referred to as *workflow application data*, some of which is also used to control the workflow execution in conjunction with the process definition. This part of application data is called *workflow relevant data*. The software components that are required to complete these tasks will be discussed in the next section.

### WfMS Typologies

According to the features of workflow, the modeling methods, the underlying technologies, and the execution modes, existing WfMS products and

prototypes can be categorized as follows[26]:

❑ *Structured* or *Ad-hoc*. For structured workflow, all the information needed to define the business process can be obtained by process analyzing and modeling prior to its actual execution. The process repeats over and over in real world. When the process definition is finished, it never or seldom changes. One example is the form processing, where a set of forms are filled and routed through a series of steps. Ad-hoc workflow has less repeatability. Some of the parameters needed to define the process may be impossible to be determined in advance and are specified at run-time. Some exceptions to normal process may also happen. The dynamic character causes many difficulties for WfMSs in modeling and execution.

❑ *Document-centric* or *Process-centric*. Documents-centric WfMSs aim to route electronic documents and/or images among multiple people for their reviewing or processing. Existing *Imaging Processing Systems* and *Document Management Systems* belong to this type. Process-centric WfMSs, on the contrary, try to model business process as a series of interdependent steps. Within each step, there are some data objects to be processed by invoking appropriate application tools, either by users themselves or by the WfMS itself. These data objects are used to construct the data objects to be passed to other steps. Many high-end WfMSs belong to this type.

❑ *Email-based* or *Database-based*. Email-based WfMSs use the e-mail system in message passing, data distributing and event notifying while executing the process instance. This method is widely adopted by many low-end systems and they run in a loosely coupled fashion. For a database-based WfMS, it stores all the data (including application data) needed in some type of DBMS. Instance execution is actually the process of retrieving and processing these data. Many high-end WfMSs that capable of managing large-scale workflow belong to this type.

❑ *Task-pushed* or *Goal-pulled*. WfMSs belonging to the former type execute activities in a process one-by-one. When one activity is finished, subsequent activities are created and activated. After all the activities have been finished, the whole process also completes. This is the execution mode implemented by most process-centric WfMSs. In a goal-pulled WfMS, a process is regarded as a goal. Upon execution, the goal is first resolved into multiple interdependent executable steps. Each of these steps can also be viewed as sub-goals that can be resolved again. When all the resolved sub-goals have been finished, the instance runs to its end point [28]. This new type of execution mode will be implemented by WfMSs of the next generation.

It should be pointed out that these typologies only characterize WfMSs from different aspects. Generally, the latter provides more flexibility than the former and will become the key feature of a WfMS that aims to manage enterprise-wide, large scale, mission-critical, complex workflow.

## WfMC Reference Model

Hundreds of WfMS or workflow-enabled products have been put into use, including ActionWorkflow by Action Technologies[16], Visual WorkFlo by FileNet[20], OpenPM by HP[25], FlowMark by IBM[22], Ultimus by Ultimus Corporation[23], InConcert by Xsoft[21] and so on. Each of these products has its own distinct features to meet specific application needs. They are implemented on different platforms with different underlying technologies, which make it very difficult for them to interoperate with each other. However, interoperability is the key to the success of enterprise wide, large scale workflow management, where different task domains are involved and different WfMSs could be used. There is a strong need for the common standards for various workflow management functions. So comes the WfMC.

Founded in Aug 1993, now with more than 200 members from both industry and research areas, WfMC aims to identify the common workflow management functional areas and develop appropriate specifications for implementation in workflow products. On Nov 29, 1994, WfMC issued its reference model[41], which completely defines the workflow, WfMS and other related concepts, the main components and the interfaces between them. Details are shown in Figure 1. The following is a brief explanation.

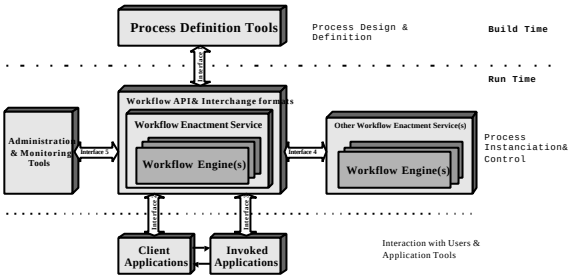


Figure 1 The WfMC Reference Model.

❑ *Process definition Tool* provides users with a tool to analyze and model actual business processes and generate corresponding computerized representations. Process definition tool interacts to the workflow enactment service through Interface 1. The interface, which is named *process definition import/export interface*[47], provides an interchange format and API calls to support exchange of process definition information.

❑ *Workflow Enactment Service* provides a run-time environment in which process instantiation and activation occurs, utilizing one or more workflow management engines, responsible for the interpreting and activating part, or all, of the process definition and interacting with the external resources necessary to process the various activities. In a situation where multiple heterogeneous workflow engines cooperate

with each other to complete the enactment of a distributed workflow, a standardized interchange is necessary between these engines. This is what Interface 4 (*Interoperability Interface*[45]) is to deal with. We will elaborate on this topic later.

- *Client Applications* let users process the tasks that need human interactions during process execution. Each of these tasks is called a *work item*, which includes some processing requirements (e.g., time limitations), the information resources and applications involved, etc. A *work list* will be maintained for every user, which contains all the work items that need to be processed by him/her. Client applications interfaces to workflow enactment services through Interface 2 (*Client Application Interface*[43,44]), which provides the APIs needed to establish sessions, to maintain processes and activities, to maintain process and activity status, to maintain worklists and to administrate process instances.
- *Invoked Applications* will be called by workflow enactment service to undertake a particular activity. Usually these applications are server-based and have no user interfaces. Information needed to access these applications, such as name, address, and parameters is part of the process definition. The Interface 3 (*Invoked Applications Interface*) defines the semantics and syntax of the APIs for standardized invocation, which include session establishment, activity management, and data handling functions. Details are under further study.
- *Administration &Monitoring Tools* are used to accomplish common administrative tasks, such as user, role and audit management, and resource control (including process instance supervisory functions and process status functions). They interact to the workflow enactment service through Interface 5(*Administration & Monitoring Interface*[42]).

The WAPI (Workflow API) is the combination of the identified five functional areas and provides a common set of API calls and related interchange formats to support them. It will be of great importance to the successful interoperability of workflow products from different vendors and the development of workflow enabled applications.

# Current Research Trends and Hotspots

The Reference Model only defines a framework according to which WfMSs may be implemented. No technical details are under consideration and they are left to the researchers and developers. As the demand for enterprise-wide, large scale, mission-critical, complex workflow management increases, WfMSs must provide more flexibility, availability, scalability, reliability, and interoperability, which, in turn, become the focuses of current research.

## More Flexible Workflow Model

WfMSs are often criticized for their "rigid" workflow models, which sometimes force users to do their work with manual coordination outside of the WfMS sphere. This is mainly because the predefined workflow model can not predicate all the cases and there are possible exceptions. GPSG[28] allows the process definitions to be incrementally refined at run-time. With GPSG, a workflow is a process grammar that specifies the lexicon of process objects (e.g., activities, documents, and roles) and the rules to combine them. A process instance is any legal phrase generated from the process grammar. Rules can be inserted, deleted or refined during instance execution. The gap between build-time and run-time functions is thus filled and much greater flexibility is achieved.

Another approach is to separate temporal and dependency relationships between activities and to mediate between abstract process definition and the concrete activity instances. FreeFlow[29] allows users to control the way in which sequencing activities are carried out in particular circumstances. To accomplish this, each task proceeds through three user states (*Inactive*, *Active*, *Ready*, which reflect potential user operations) and three system states (*Disabled*, *Enabled*, *Pending*, which record dependencies between activities). FreeFlow is then to maintain correspondence between the two sets of states.

## Object-oriented WfMSs

Object-oriented technology has been widely used in research and application areas. Researches on object-oriented WfMSs (OOWfMS) include two areas. The first is to implement WfMS with some object-oriented programming method, which models various elements in the WfMS as objects and implements them on some platform. Examples include InConcert[37], FlowMark[11], CodAlf[2], Wowww![39], etc. Another more interesting area is to model business process with object-oriented analysis technology and to construct object-oriented process model[3,30]. The key to the success of OOWfMS is to achieve high level of reusability of the process definition.

## Intelligent WfMSs

It has been recognized that artificial intelligence (AI) will be of great importance to the success of groupware systems[5]. This is because AI approach is usually heuristic or augmentative, which allows information to accrue through user-machine interaction rather than being initially complete and structured. Modeling business processes is a very onerous task for modelers. At the same time, because not all parameters can be determined in advance, the resulting process definition is sometimes incomplete. An intelligent WfMSs with self-learning capability will be able to capture the information needed to construct or complete process definitions automatically during enacting[21]. In a goal-pulled WfMS, various searching algorithms can be used to

determine the sub-tasks needed and the sequencing of them for the successful completion of a certain high-level task.

## Support for Synchronous Cooperation

WfMSs have long been viewed as an asynchronous groupware system, where support for synchronous cooperation is unavailable. However, some tasks in a large workflow may need real-time cooperation of multiple users to get done. For example, process modeling can be finished more efficiently with the support of a cooperative process definition tool. In a traditional WfMS, a work item can be processed by one only user, which can impose great limitations on the support for certain types of tasks, e.g., cooperative design[35], cooperative editing[6], and conferencing[49]. Wowww!'s[39] solution is to provide synchronous application tools (e.g. a white board) embedded in its application form and to allow multiple users to access the same work item (represented as an application form) in a synchronous mode. However this support raises many other issues in consistency control, session maintenance, and process definition. Other alternatives include integrating synchronous multimedia collaboration into workflow management[13].

## Support for Mobile Users

WfMSs provide an ideal software platform for mobile office. However, most early WfMSs are developed in a LAN environment, where users maintain uninterrupted connections to the system. In a mobile computing environment where only low rate, unreliable, yet more expensive network connections are available, users have to download the work items to be processed to local machines and then disconnect[38]. When the task is finished (perhaps several hours or days later), the connection is reestablished and the result is submitted to the system. A protocol between client applications and workflow engine is thus required A good protocol should provide users with full autonomy and maintain data consistency (e.g., preventing a work item from being processed by multiple users). One approach is to let the modelers specify which tasks can be processed in a disconnected mode[7]. Other alternatives include defining a three-phase protocol (prior to disconnecting, disconnected and reconnecting)[12].

## Web-based WfMSs

WWW provides an easy to use and platform independent interface to access various heterogeneous resources (e.g., files, databases, e-mails, etc.) that have been deployed in the Internet/Intranet environment. Web-based WfMS[9,48] is a newly emerged research trend that tries to integrate those resources in activity coordination and to present information specific to the WfMS (e.g., worklists, status of process instances, and the information of the task to be processed) through HTML pages generated by CGI programs that interface to the workflow engine. However, HTML lacks to power to develop more interactive applications and to process more complex data objects. The low efficiency of CGI programs can also result in bottlenecks. With the maturity of Java technology, these shortages can be overcome[39].

## Distributed WfMSs

Usually, a WfMS uses only one workflow engine, which can result in serious single site failures and bottlenecks. Distributed WfMSs try to enact process instances through the cooperation of multiple engines distributed in a network environment to increase the reliability, scalability and availability. Each engine is responsible for the execution of different instances or different parts of an instance. When one engine crashes or becomes busy, the work can be transferred to other engines. One approach used by Ultimus is to group multiple workflow servers as a "virtual server", another approach by FlowMark is to use clustered workflow servers connected to the same object store. However, distributed architecture raises more difficulties in the maintaining the status of process instances and the worklists. Since almost all WfMSs use some type of DBMS to store various data involved in instance execution, some ideas from distributed data management area, e.g., data replication, distributed query optimization, and recovery can also be introduced into distributed workflow architecture.

## Transactional WfMSs

Researches on transactional WfMSs try to apply some advanced transaction models (e.g., Saga[15], Flexible Transaction[1], etc.) to instance execution to increase the reliability, e.g., to ensure the integrity and consistency of activities in a process instance[10,33]. When failure occurs, the process instance should be rolled back to a semantically acceptable state. However, some concepts in transaction processing, such as ACID properties and failure semantics still remain unclear in workflow domains. Existing results show that workflow model has much richer semantics than advanced transaction model and provide an ideal environment to implement these models[8,43]. One important aspect in transactional WfMS research is process compensation, which means that how to undo the effects of a failed or canceled activity. The key is to reduce the number of activities that should be compensated and re-executed (compensation scope), since these operations can be quite expensive. According to the data dependencies among activities, strategies used to determine the scope have been proposed[40]. Dependencies can also be maintained as a basic service during instance execution[32].

### Interconnecting Heterogeneous WfMSs

Large-scale workflow management requires various WfMSs with distinct features to be interconnected so that complex business processes that involve many task domains can be coordinated through the cooperation of these WfMSs. In a distributed WfMS, there is also the need to interconnect multiple workflow servers. WfMS interconnection includes three key aspects. Firstly, the interoperation model, which refers to the relationship between two process instances enacted in two workflow servers. Identified models include *chained*, *nested-subprocess*, *pear-to-pear*, and *parallel synchronized*. Secondly, the level of interoperability, which refers to the extent to which the interoperability is achieved. Eight levels have been defined, from the simplest *no interoperability* to the most complex *common look and feel utilities*. Thirdly, according to the model and level chosen, an *interoperation set* (APIs that allow the communication between different servers) should be defined. The interoperability solution proposed by WfMC, which adopts a nested model and achieves level 4 interoperability, uses Internet e-mail to send information between workflow servers[46]. However this loosely coupled fashion may not be appropriate for some processes and management tasks. So more efficient protocols are required.

Another aspects in workflow research that have not been addressed in this paper due to limited pages include integration with Business Process Re-engineering and workflow benchmarks. Interested readers are referred to [18,19,24].

## Concluding Remarks

This paper summarizes the current state of the researches on workflow management. We believe that WfMS will contribute a lot to the success of an organization in the next century and will become one kernel component of organization's information infrastructure.

## Acknowledgements

This project is supported by China National Advanced Science & Technology (863) Plan under contract NO. 863-306-ZD-10-2B. The authors would like to thank other members in the Computer Network and CSCW Research Lab. for their support during the conduct of this project.

## References

1 A.K. Elmagarmid, Y. Leu, W. Litwin, and etc. *A Multidatabase transaction model for interbase.* Proc of VLDB Conf'90, 1990

2 Alexander Schill, Christian Mittasch. *CodAlf: A decentralized workflow management system on top of OSF DCE and DC plus plus.* Proc Of Int'l Symposium on Autonomous Decentralized Systems'97, 1997:205-212

3 Atsushi Inamoto. *Object-oriented model driven workflow process analysis.* Proc Of Japan/USA Symposium on Flexible Automation'96, 1996(2):1317-1324

4 C. A. Ellis, G. J. Nutt. *Modeling and enactment of workflow systems.* Application and Theory of Petri Nets, Springer-Verlag, Chicago, III, 1994.

5 C. A. Ellis, S. J. Gibbs, G. L. Rein. *Groupware: some issues and experiences.* Comm Of ACM, 1991'34(1):39-58

6 C. A. Ellis, S. J. Gibbs. *Concurrency control in groupware systems.* Proc ACM SIGMOD'89: 399-407.

7 C. Bussler. *User Mobility in workflow-management systesm.* Proc Of Telecommunications Information Networking Conference'95, 1995

8 C. Mohan, G. Alonso, R. Gunthor, and etc. *An overview of the Exotica research project on workflow mnagement systems.* Proc Of IEEE Int'l workshop on high performance transaction systems'95. 1995

9 Charles K. Ames, Scott C. Burleigh, Stephen J. Mitchell. *A web based enterprise workflow system.* Proc Of ACM SIGGROUP'97, 1997:214-220

10 G. Alonso, D. Agrawal, A. El Abbadi, and etc. *Advanced transaction models in workflow contexts.* Proc Of IEEE Int'l Conf. on data engineering'96,1996: 574-581

11 G. Alonso, M. Kamath, Dl Agrawal, and etc. *Failure handling in large-scale workflow management systems.* IBM Research Report RJ9913, November 1994.

12 G. Alonso, R. Gunthor, M. Kamath, and etc. *Exotica/FMDC: handling disconnected clients in a workflow management system.* Proc Of Int'l Conf on Cooperative Information systems'95, 1995

13 Georg Schneider, Michael Weber, Gerhard Partsch, etc. *Integrating synchronous multimedia collaboration into workflow management.* Proc of ACM GROUP'97, 1997:281-290

14 Gustavo Alonso, Berthold Reinwald, C. Mohan. *Distributed data management in workflow environments.* Proc Of IEEE Int'l workshop on research issues in data engineering'97. 1997:82-90

15 H. Garcia-Molina, K. Salem. *Sagas.* Proc Of SIGMOD'87, 1987:249-259

16 http://www.actiontech.com/

17 http://www.aiim.org/wfmc/

18 http://www.cbl.ncsu.edu/

19 http://www.doculabs.com/

20 http://www.filenet.com/

21 http://www.inconcertsw.com/

22 http://www.software.ibm.com/ad/flowmark/
23 http://www.ultimus1.com/
24 http://www.waria.com/waria/
25 J. Davis, W. Du, M. Shan. *OpenPM: An enterprise process management system.* IEEE Data Engineering Bulletin, 1995.
26 Kenneth R. Abbott, Sunil K. Sarin. *Experiences with Workflow Management: Issues for the Next Generation.* Proc of CSCW'94 , 1994:113-120
27 M. Satyanarayanan. *Fundamental Challenges in Mobile Computing.* Proc of ACM Symposium on PODC'96, 1996:1-7
28 Natalie S. Glance, Daniele S. Pagani, and Remo Pareschi. *Generalized process structure grammars(GPSG) for flexible representations of work.* Proc of CSCW'96, 1996:180-189
29 Paul Dourish, Jim Holmes, Allan MacLean, and etc. *Freeflow: mediating between representation and action in workflow systems.* Proc of CSCW'96, 1996:190-198
30 Peter C. Lockemann, Hans-Dirk Walter. *Object-oriented protocol hierarchies for distributed workflow systems.* Theory and Practice of Object Systems, 1995'1(4):281-300
31 Raul Medina-Mora, Terry Winograd, Rodrigo Flores, and etc. *The Action workflow approach to workflow management technology.* Proc of CSCW'92, 1992: 281-288
32 Roger Guenthoer. *Dependency manager a base service for transactional workflow management.* Proc of IEEE Int'l workshop on research issues in data engineering'96, 1996:86-95
33 Rusinkiewicz A. Sheth. *On Transactional workflows.* Bulletin of the technical committee on data engineering, 1993'16(2)
34 Shi Meilin. *CSCW：Computer Supported Cooperative Work*. *China Journal of Communications*, 1995'16(1):55~61
35 Shi Meilin, Yang Guangxin. *CoDesign: Prototype of a Cooperative Design Support System.* Journal of Tsinghua University, 1998'38(S1)
36 Stefano Ceri, Paul Grefen and Gabriel Sanchez. *WIDE - a distributed architecture for workflow management.* Proc of IEEE Intl workshop on research issues in data engineering'97, 1997:76-79
37 Sunil K. Sarin. *Object-oriented workflow technology in InConcert.* Proc of IEEE Computer Society Int'l Conf'96, 1996:446-450
38 T. Imielinski, B. R. Badrinath. *Mobile wireless computing: solutions and challenges in data management.* Comm of ACM, 1994'37(10)
39 Yang Guangxin, Shi Meilin, Xiang Yong, etc. *Wowww!: managing workflow on the world wide web.* Proc of ICCT'98, Beijing, 1998
40 Weimin Du, Jim Davis, Ming-Chien Shan. *Flexible specification of workflow compensation scopes.* Proc of ACM GROUP'97, 1997:309-316
41 WfMC. *The workflow reference model.* 1994
42 WfMC. *Workflow management coalition audit data specification.* 1996
43 WfMC. *Workflow Management Coalition Workflow Client Application (Interface 2) Application Programming Interface (WAPI) Specification.* 1996
44 WfMC. *Workflow Management Coalition Workflow Client Application (Interface 2) Application Programming Interface (WAPI) Naming Conventions.* 1996
45 WfMC. *Workflow management coalition workflow standard - interoperability abstract specification.* 1996
46 WfMC. *Workflow Management Coalition workflow standard – Interoperability Internet e-mail MIME binding.* 1996
47 WfMC. *Workflow process definition read/write interface: request for comments.* 1995
48 Wolfgang Gräther, Wolfgang Prinz and Sabine Kolvenbach. *Enhancing workflow by web technology.* Proc of ACM GROUP'97, 1997:271-280
49 Wu Shangguang, Shi Meilin. *Support environment for CSCW research - design and implementation of a desktop computer conferencing system.* Proc of ICCT'96 Beijing, 1996:187-190