A Taxonomy of Workflow Architectures

Kwang-Hoon Kim* · Su-Ki Paik*

⟨Abstract⟩

This paper proposes a conceptual taxonomy of architectures for workflow management systems. The systematic classification work is based on a framework for workflow architectures. The framework, consisting of generic-level, conceptual-level and implementation-level architectures, provides common architectural principles for designing a workflow management system. We define the taxonomy by considering the possibilities for centralization or distribution of data, control, and execution. That is, we take into account three criteria. How are the major components of a workflow model and system, like activities, roles, actors, and workcases, concretized in workflow architecture? Which of the components is represented as software modules of the workflow architecture? And how are they configured and operating in the architecture?

The workflow components might be embodied, as active (processes or threads) modules or as passive (data) modules, in the software architecture of a workflow management system. One or combinations of the components might become software modules in the software architecture. Finally, they might be centralized or distributed. The distribution of the components should be broken into three: Vertically, Horizontally and Fully distributed. Through the combination of these aspects, we can conceptually generate about 64 software Architectures for a workflow management system. That is, it should be possible to comprehend and characterize all kinds of software architectures for workflow management systems including the current existing systems as well as future systems.

We believe that this taxonomy is a significant contribution because it adds clarity, completeness, and global perspective to workflow architectural discussions. The vocabulary suggested here includes workflow levels and aspects, allowing very different architectures to be discussed, compared, and contrasted. Added clarity is obtained because similar architectures from different vendors that used different terminology and techniques can now be seen to be identical at the higher level. Much of the complexity can be removed by thinking of workflow systems. Therefore, it is used to categorize existing workflow architectures and suggest a plethora of new workflow architectures.

Finally, the taxonomy can be used for sorting out gems and stones amongst the architectures possibly generated. Thus, it might be a guideline not only for characterizing the existing workflow management systems, but also for solving the long-term and short-term architectural research issues, such as dynamic changes in workflow, transactional workflow, dynamically evolving workflow, large-scale workflow, etc., that have been proposed in the literature.

Key words: *Software Architecture, Workflow Management Systems, Workflow Enactment Architectures*

* Department of Computer Science, Kyonggi University

1. Introduction

The software architecture level of design in software development processes has been becoming an important step when designing complex software. So far, there have been several well-defined terminology and notation to characterize architectural structures in the software development fields[8]. In the meantime, there has been an implicit body of work on the software architecture in research and development fields of the workflow management systems and technologies. Usually, it is used to only describe systems. So, there is not a systemically well-defined terminology, notation or classification to characterize architectural structures in the workflow literature. Almost all efforts on this topic have been concentrated on the workflow enactment part of workflow management systems, which is capable to create, manage and execute workflow procedure's workcases. Many of them represent only the implementation level details[2,3,11]. Others are simply documented at the conceptual level details[10,12,15,16]. In short, there are not well-defined criteria or common senses in terms of defining and describing the software architecture of a workflow management system.

This paper tries to suggests a conceptual-level taxonomy of workflow architectures to grasp all kinds of software architectures for workflow management systems including the current existing systems as well as future systems supporting highly advanced workflow features, like dynamic changes, transactional workflow management and large scale workflow management. After all, it might play an important role in guiding for designing and developing a workflow management system. Because these works on taxonomy are closely related with how the enactment control of workflow is configured, designed and implemented.

In the following sections, we describe criteria for this taxonomy work. Next, we suggest classified workflow architectures based on the taxonomy and classify the current workflow management systems. At the end of this paper, we sort gems and stones out of the classified architectures so as to figure out which architectures fit well into which advanced workflow requirements, like scalability, availability, and extensibility, for large scale workflow, dynamic changes in workflow, transactional workflow and groupware support in workflow.

2. Criteria for Taxonomy

There are several criteria being used to characterize architectures in terms of conceptual aspects. We have sought the optimal set of criteria coping with grouping and characterizing architectures.

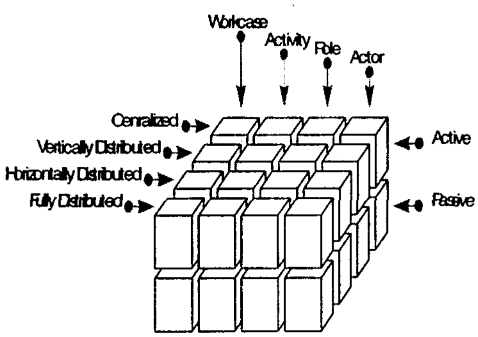

Figure 1: Three Dimensional Criteri

Figure 1 illustrates the finally decided criteria, which are represented by three dimensions, for a conceptual level taxonomy of all possible architectures for the existing workflow management systems as well as for future workflow management systems:

- *a pattern of enactment control* concerning how to schedule the execution of activities,
- *a type of workflow model* concerning which kind of workflow models is used for managing and operating workflow management.
- *a type of process structure* concerning how each workflow component is concretized in an architecture.

So, it is possible to classify workflow architectures through the combinations of these three criteria. The following is to describe the concepts of and the relationships among these criteria.

Centralized Control vs. Distributed Control

A pattern of distribution structure for the workflow enactment control and schedule must be able to become one of criteria for the conceptual level taxonomy of workflow architectures. Because there must be a software module, playing a role of workflow enactment control and schedule, in a workflow management system. Roughly speaking, it is not too much to say that the software module discharges almost all functions of a workflow management system. The software component (scheduler) is able to perform the scheduling work in a different fashion.

The first is that the scheduling work of the activity enactment gets done through a single performer. The second is that the work is done by multiple performers which have the same functionality. So, each performer takes over a portion of workcases of a procedure and has the responsibility of enactment control for them. The third is that the work is completed by the collaborative work of several performers, each of which has a different functionality. That is, this is to allot a portion of the work to each performer. So, each of the performers fulfills its own role in the scheduling work for all workcases of a procedure. The final is that the work is done by the combined fashion of both the second and the third. These scheduling fashions are called the centralized, the vertically distributed, the horizontally distributed and the fully distributed workflow enactment scheduling mechanism, respectively.

Workcase (Procedure), Activity, Role vs. Actor

A workflow model is used for building a workflow procedure through the main components: activities, roles, and actors. Additionally, there is another component, which is called workcases. The workcase is an instance or workload of a workflow procedure. But, we advocate an open-ended list of possible components. The list is something that may expand in the future as workflow systems encompass social and organizational aspects more comprehensively. A list of components could, for example, include a temporal component, an information component, a social component, a resource component, and others. In this paper, we consider only major components identified by the Workflow Management Coalition, and defined in standard workflow definition[17]. Thus, the major components in our conceptual workflow model are workcases, activities, roles and actors.

Based on the components, four kinds of workflow model are available: Workcase-based, Activity-based, Role-based and Actor-based workflow model. That is, it's depending on how a workflow procedure is represented in a workflow management system. Then, the workflow management system is going to operate based on the workflow procedure information organized by the component (workflow model).

Active-Model vs. Passive-Model

The type of process structure consists of an active process structure type and a passive process structure type. In the conceptual architectures, it is so crucial how the workflow models are transformed or embodied onto architecture. In other words, they can be alive as an active component in architecture. Or, they can just be an passive component, like data stored on database. The former is called *active-model*, the later is called *passive-model*. Suggestively speaking, almost of the current commercialized workflow management systems are based on the passive-model. It is possible to compare their properties to each other as following:

- The configuration of a conceptual architecture is flexible in an active-model, meanwhile it is firm in a passive-model.
- While, in an active-model, the architecture is reflective of the structures of workflow procedures, in a passive-model, architectural components are organized by taking its functionality into consideration.
- For an instance, a workflow procedure is realized, as a process or a thread, in a active-model workflow management architecture. While on the other, in the passive-model workflow management architecture, it is stored as data.

Class-Active vs. Instance-Active

Again, the active-model concept should be divided into three sub-concepts: class-active, instance-active and hybrid, according to how workcases from a procedure are reflected in architecture. In the class-active case, workcases are represented as a passive component. Workcases are becoming active components of architecture in the instance-active case. The hybrid case is to combine both the class-active and the instance-active. The detailed explanation about these active-models and the typical architectures acting for each of the cases will be presented in the next section.

3. Taxonomy of Architectures

Putting all criteria together, we are able to generate several types of workflow architectures. At the top level of the taxonomy, there are three categories of workflow architectures: *Passive-model*, *Class-active* and *Instance-active* workflow architectures. The next level of the taxonomy is the workflow model taken by workflow management systems. The representation of the workflow model is completely different in the active-model from in the passive-model workflow architecture. So, the taxonomy is again subdivided into five categories:

Workcase-based, Activity-based, Role-based, Actor-based and *Hybrid* workflow architectures. The final level of the taxonomy is the pattern of distribution structure for the workflow enactment control and schedule: *Centralized, Horizontally-Distributed, Vertically-Distributed* and *Fully-Distributed* workflow architectures. Also, the distribution structures are able to be differently realized in the active-model and the passive model workflow architecture. In the following sections, detailed explanations about these classified workflow architectures are presented.

In general, there should be four kinds of software modules on the workflow architecture: workflow enactment scheduling and event handling server, workflow client, invoked applications, and workflow information manager. The taxonomy focuses on how to design and implement the workflow enactment scheduling and event handling server, which is the most important module in the workflow architecture, by considering the three dimensions of criteria.

3.1 Passive-Model Architectures

So far, several hundreds workflow management systems have been developed and commercialized. In terms of their workflow architecture, almost all are classified into the passive-model workflow architecture, because their software architecture is based on the concept of the passive-model in terms of the process structure for the workflow model.

The workflow models use to be transformed into data so as to be managed by the workflow information manager. The process structures are related with how to organize the workflow enactment scheduling and event handling server. As a necessary consequence, the passive-model architecture is subdivided into 20 sets by the combination of the workflow model and the distribution structure.

In the passive-model workflow architecture, the distribution structures of the workflow scheduling and event handling server are realized as following:

- Centralized: There is a server fully discharging the workflow enactment control and schedule for all workcases out of a procedure. Almost all workflow management systems currently available in the market belong to this category of the distribution structure.
- Vertically distributed: There are multiple copies of a server taking fully responsibility for the work. So, a portion of workcases being produced from a procedure is assigned into each of them. In this case, they need not to cooperate or communicate each other to do the work.
- Horizontally distributed: There are several servers taking partial charge of the work. So each sever has an exclusive responsibility on its allotted task out of the workflow enactment control and schedule work for all workcases from a procedure. There should be a kind of cooperation and communication among the servers to accomplish the work.
- Fully distributed: In this case, workcases belonging to a procedure are handled by a combined fashion of the vertical and horizontal distribution. That is, multiple copies of servers are going to be exclusively discharging

their own allotted tasks which are divided from the workflow enactment control and schedule work. As a result, a portion of workcases should be assigned into a set of servers needed to complete the scheduling work.

3.2 Class-Active Architectures

In recent, the distributed pattern of control structure has been issued in the workflow literature. The distributed enactment control structure can be naturally implemented in the active-model workflow architecture. In this section, we show how the workflow model and the distribution become operational in the class-active workflow architecture.

In the class-active workflow architecture, the workflow model becomes an active module according to one of the following alternatives:

- Procedure-based embodiment: whenever a procedure is created, it becomes an active software module.
- Activity-based embodiment: whenever a procedure is created, it is represented into a number of active components corresponding to the activities being included in itself.
- Role-based embodiment: whenever a procedure is created, it is represented into a number of active components corresponding to the roles being associated within itself.
- Actor-based embodiment: whenever a procedure is created, it is represented into a number of active components corresponding to the actors being associated with itself.
- Hybrid embodiment: whenever a procedure is created, it is represented into a number of active components corresponding to any combinations of four components.

Also, each set of active components, which is associated with a workflow procedure, operates according to the distribution structures realized as one of the following alternatives:

- Centralized: This is the case of that a procedure or its workcases perform the workflow enactment controlling and scheduling work. In this case, a workcase-based active-model component in an architecture is used to be fully discharging all about the corresponding procedure such as monitoring, enactment controlling and scheduling for all workcases out of the procedure. Therefore, one workcase-based active-model component must be created whenever a procedure is defined in a workflow management system.
- Vertically distributed: There are multiple copies of the workcase-based active-model component, corresponding to a procedure, in an architecture. So, a portion of workcases being produced from the procedure is assigned into each of them. In this case, they need not to cooperate or communicate each other to do the work.
- Horizontally distributed: This is the case of that the components but procedures and workcases, such as activities, roles and/or actors, of the workflow model have the work controlling and scheduling the workflow enactment. For an example, an activity-based active-model component may

be exclusively responsible on all about the corresponding activity. Therefore, there must be a kind of cooperation and communication among those components to accomplish the work.

- Fully distributed: This is a case of that an architecture is configured with a combined fashion of the vertical and horizontal distribution.

3.3 Instance-Active Architectures

Based on our literature survey, there has not been any proposed or implemented workflow architecture that possesses the properties of the instance-active architecture. We believe that the instance-active workflow architecture should be very appropriate for maximizing distribution capability in the enactment scheduling and event handling works. Because the subject of the work, which is becoming a software module, is a workflow procedure in the class-active architecture, while on the other it is workcases associated with a workflow procedure in the instance-active architecture. In this section, we describe about the embodiment of the workflow model in the instance-active workflow architecture.

- Workcase-based embodiment: whenever a workcase is created, it becomes an active component.
- Activity-based embodiment: whenever a workcase is created, it is represented into a number of active components corresponding to the activities being included in itself.
- Role-based embodiment: whenever a workcase is created, it is represented into a number of active components corresponding to the roles being associated within itself.
- Actor-based embodiment: whenever a workcase is created, it is represented into a number of active components corresponding to the actors being associated with itself.
- Hybrid embodiment: whenever a workcase is created, it is represented into a number of active components corresponding to any combinations of four components.

The operation of the distribution in the instance-active and hybrid workflow architecture is similar to in the class-active workflow architecture.

3.4 Taxonomy of Current WFMSs

In recent, quite a few workflow management systems have been developed and commercialized in the workflow literature. Figure 2 presents the architectural taxonomy for some of those current available workflow management systems. In the taxonomy, only two dimensions, the process structure and the pattern of workflow enactment control, are considered, because the dimension of the model's component is a little effective in the passive-model conceptual architecture on which the components are represented as data. In other words, the dimension becomes effective in doing taxonomy for the active-model conceptual architectures. But, there have been proposed a few workflow management systems professing the active-model workflow architecture.

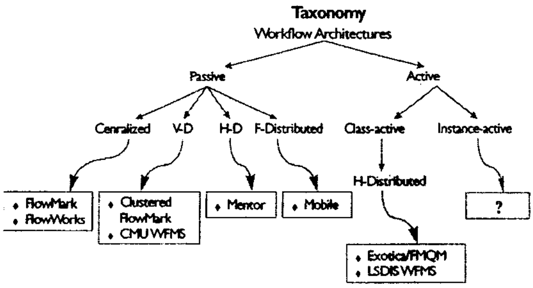

Figure 2: Taxonomy of Current WFMSs

Almost all commercialized products, such as InConcert, Staffware, FloWare, Notes, ActionWKF, FlowWorks and FlowMark workflow management systems, are adopting a platform of the client-server architecture with a database management system. It may safely be said that those systems are categorized into the passive-model and centralized conceptual architecture. Especially, the FlowWorks workflow management system[1] made by BULL and the FlowMark workflow management system [2,3,12] produced by IBM are typical examples for the conceptual architecture of passive-model, centralized enactment control pattern and workcase-based representation.

There is a good example for the passive-model, vertically distributed and workcase-based conceptual workflow architecture. D. Alonso and et al[11] at IBM proposed an extended version of the FlowMark workflow management system, which is originally designed to enhance availability of the workflow system by handling failures. That is called *clustered server architecture* for FlowMark. The architecture consists of several clusters, each of them with its own database on which the same workflow procedure information is stored, so that workcases out of a workflow procedure are disseminated into and handled by these clusters.

CMU's static and dynamic workflow architectures developed by K. Wallnau and et al[12] are another examples of the passive-model, vertically distributed and workcase-based conceptual workflow architecture. Because one of the key ideas expressed in the architecture is executing workflow engines represent the instantiation of a workflow procedure description (model), and a single workflow procedure may be simultaneously instantiated by several workflow engines.

There is a typical workflow management system being fitted into the passive-model and horizontally distributed workflow conceptual architecture. That is the mentor workflow management system's architecture proposed by Jeanine W. et al[10]. It is designed so as to allow for diversity and for scalability by an approach in which a workflow procedure can be partitioned into a number of subworkflows, each of which is handled by a different server.

The typical workflow management system belonging to the passive-model and fully-distirbuted workflow conceptual architecture is the *mobile* workflow management system that has been proposed by Stefan J.[16]. It supports the specification of workflows through several modeling perspectives, such as functional, behavioral, informational, operational and organizational perspectives. Each of the perspectives is represented into a passive component and handled by the corresponding server on the architecture. And servers corresponding to the perspectives are able to be replicated according to the scale of workflow procedures and their workcases.

According to our literature survey, there have been two proposed workflow management systems professing the class-active workflow architecture. These are just Exotica/FMQM, a persistent message-based architecture for distributed workflow management by IBM[3], and LSDIS's workflow architectures, CORBA-based enactment architectures for workflow management systems by the large scale distributed information systems Lab At the university of Georgia.

In the Exotica/FMQM persistent message-based architecture, a workflow procedure is divided by several subworkflows, each of which is concretized as an active components in the architecture, and the active components are located into several nodes in network. Each of the active components will be in charge of coordinating the execution of workcases out of the corresponding subworkflow by examining the queue for incoming messages and creating an activity thread to manage the execution of the activity. Therefore, this workflow management system is a typical example of the class-active-model, horizontally distributed and active-based conceptual workflow architecture.

A.P. Sheth and his colleague[15] at LSDIS Lab have proposed five conceptual workflow enactment architectures for implementing a work-flow management system. The fully-distributed workflow architecture among them is an another example for the class-active-model, horizontally distributed and activity-based conceptual workflow architecture. The activity manager in the architecture houses an individual scheduler without a centralized scheduler and controls the enactment of workcases through asynchronous IDL interfaces providing the communications among activity managers.

4. Architectures for Future WFMSS

In this section, we are trying to match the categories of the taxonomy into some of architectural requirements[14] for future workflow management systems. Figure 3 shows the results.

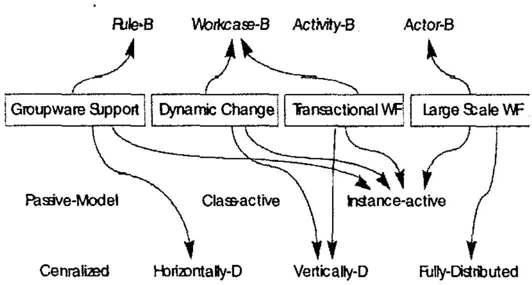

Figure 3: Architectural Properties of Future WFMSs

Merging Workflow with Groupware

Groupware has been defined as "technology based systems that support groups of participants working on a common task or goal, and help provide a shared environment". It naturally includes technologies such as electronic mail, video conferencing, and shared document editors. The original motivation of this requirement is well specified in[6].

If one workflow architecture is designed by the instance-active, role-based workflow model and horizontally distributed enactment scheduling and event handling pattern, we can expect better efficiency in supporting groupware applications in workflow, when considering the characteristics of groupware applications and technology.

Handling Exceptions: Dynamic Changes

The workflow management system is an efficient office information system, at the same time, a most fragile system if it is not equipped by robust exception handing mechanisms, because it manages and controls the flow of works in office in where several kinds of exceptions must be highly happened[6].

For this requirement, the workflow architecture might have the instance-active, workcase-based workflow model and vertically distributed control pattern.

Transactional Workflow

The motivation of this requirement[9,14] is in correctness and reliability of applications implementing business or information processes. That is, when multiple objects such as databases, files, documents, devices are accessed by a workflow execution, data consistency problems can arise either from concurrency, application failures, system failures, or network failures. These introduce the need for concurrency control, recovery, and transaction coordination in workflow systems. Most commercial workflow management systems provide limited capabilities to deal with these problems.

The characteristics of transactions are similar to the instance-active, workcase-based workflow model and vertically distributed control pattern. So, one might be expected better efficiency in the transactional workflow applications by taking the pattern.

Large Scale Workflow

Workflow architecture is the execution infrastructure for workflows which is deeply related with performance issues such as reliability, scalability, extensibility, and robustness. Most of commercial workflow management systems have the server-client architecture. Finally, the server might become a bottleneck of the performance in larger scale application domains[3,4,14].

For this requirement, the control of workflow should be maximally distributed and based on actors belonging to geographically distributed organizations. So, the instance-active, actor-based workflow model and fully distributed pattern might be suitable for large-scale workflow architectures.

5. Conclusions

So far, the conceptual-level taxonomy for workflow architectures has been done according to the three dimensions of criteria. And, based on the taxonomy, we tried to classify the current workflow management systems. We believe that the active-model architectures, especially the instance-active architecture, should be more appropriate for satisfying the advanced workflow requirements to solve long-term and short-term architectural research issues. At the same time, the taxonomy is allowing very different architectures to be discussed, compared, and contrasted. Also, it is used to categorize existing workflow architectures and suggest a plethora of new workflow architectures.

As further research issue, we are going to design and implement workflow architectures that are professing the instance-active architecture and supporting the advanced workflow requirements.

〈References〉

[1] "FlowWorks, The Bull Workflow Product: Architectural Design and Functional Specification", BULL L.P.M., Oct. 1991

[2] Alonso, D. Agrawal, A. El Abbadi, M. Kamath, R. Gunthor and C. Mohan, "Advanced Transaction Models in Workflow Contexts", IBM Research Report, 1997

[3] Alonso, D. Agrawal, A. El Abbadi, M. Kamath, R. Gunthor and C. Mohan, "Exotica/FMQM: A Persistent Message-Based Architecture for Distributed Workflow Management", Proceedings of IFIP Working Conference on Information Systems for Decentralized Organizations, Aug. 1995

[4] Alonso, D. Agrawal, A. El Abbadi, M. Kamath, R. Gunthor and C. Mohan, "Failure Handling in Large Scale Workflow Management Systems", IBM Research Report RJ9913, Nov. 1994

[5] Clarence A. Ellis and Marc Bernal, "Officetalk-D: An Experimental Office Information System", Technical Report, Xerox Palo Alto Research Center, 1987

[6] Clarence A. Ellis and Carlos Maltzahn, "Chautauqua: Merging Workflow and Groupware", Technical Notes, Department of Computer Science, June 1996

[7] Clarence A. Ellis and Gary J. Nutt, "Workflow: The Process Spectrum", Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions, May 1996

[8] Dewayne E. Perry, Alexander L. Wolf, "Foundations for the Study of Software Architecture", ACM SIGSOFT Software Engineering Notes, Vol. 17, No. 4, Oct 1992

[9] Dimitrios Georgakopoulos and Mark F. Hornick, "A Framework for Enforceable Specification of Extended Transaction Models and Transactional Workflows", International Journal of Intelligent and Cooperative Information Systems", Vol. 3, No. 3, pp. 225-253, 1994

[10] Jeanine Weissenfels, Dirk Wodtke, Gerhard Weikum and Angelika Kotz Dittrich, "An Overview of the Mentor Architecture for Enterprise-wide Workflow Management", Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions, May 1996

[11] Kamath, G. Alonso, R. Gunthor and C. Mohan, "Providing High Availability in Very Large Workflow Management Systems", 5th International Conference on Extending Database Technology, March 1996

[12] Kurt Wallnau, Fred Long and Anthony Earl, "Toward a Distributed, Mediated Architecture for Enterprise-wide Workflow Management", Proceedings of NSF Workshop on Workflow and Process Automation in Information Systems: State- of-the-Art and Future Directions, May 1996

[13] Kwang-Hoon Kim, "Practical Experience on Workflow: Hiring Process Automation by FlowMark", IBM Internship Report, IBM/ISSC Boulder Colorado, 1996

[14] Kwang-Hoon Kim, Su-Ki Paik, "Practical Experiences and Requirements on Workflow", Lecture Notes Asian '96 Post-Conference Workshop: Coordination Technology for Collaborative Applications,

The 2nd Asian Computer Science Conference, Singapore, 1996

[15] Miller, A.P. Sheth, K.J. Kochut and X. Wang, "CORBA-Based Run-Time Architectures for Workflow Management Systems", Large Scale Distributed Information Lab. Report, University of Georgia, 1996

[16] Stefan Jablonski and Christoph Bussler, "Workflow Management: Modeling Concepts, Architectures and Implementation", International Thomson Computer Press, 1996

[17] Jari V., et al, "Research Issues in Workflow Systems", VTT Information Technology Report, Oct. 1995