Impact of Multimedia Data on Workflows*

Aidong Zhang
Department of Computer Science
State University of New York at Buffalo
Buffalo, NY 14260 USA

Abstract

Workflow has been used to manipulate the relevant tasks within a group activity in a cooperative environment. Such cooperative work usually demands the creation and exchange of information by means of a wide variety of media, resulting in multimedia workflows. Spatiotemporal dependencies among tasks represent those new relationships that appear in the multimedia environment. In this paper, we discuss the effects of multimedia data on workflow specification and management in multimedia database systems. Two major types of the spatiotemporal dependencies among tasks, namely, progressive parallel and real-time precedence dependencies, are identified. These spatiotemporal dependencies propose new challenges to the workflow management in multimedia database systems. A Multimedia Workflow Specification and Management Environment (WSME) is to be developed that supports the definition and execution of application-specific multimedia workflows. The WSME provides a multimedia workflow specification language that allows users to define multimedia workflows. In addition, the WSME provides multimedia workflow management mechanisms that maintain synchronized retrievals in a context of multiple data streams. The combination and presentation of continuous data streams in different media, such as images and speech fragments, can only be achieved through a careful synchronization of the ordering of these data streams in the time domain.

1 Introduction

Computer-supported cooperative work (CSCW) has recently been a very active area for research, development, and production, particularly in distributed environments [IEE94]. Within this area, research has been proposed to clearly specify the control and data flow of cooperative activities, which is generally termed *workflow*. A workflow model uses a declarative control and data flow specification language to facilitate a separation between the application code on one hand, and the control and data flow among tasks of the application on the other [BDS+93]. Various intertask dependencies have been recognized.

*This paper is presented in the CSCW-94 Workshop on Distributed Systems, Multimedia, and Infrastructure Support in CSCW.

The need to incorporate multimedia data in database systems is growing rapidly in many fields, including business, manufacturing, education, Computer Aided Design (CAD), and medicine. People naturally create and exchange information by means of a wide variety of media. They talk to each other, draw pictures, label diagrams, and write notes. Accordingly, multimedia data are generally classified into five categories: images, audio, video, text, and numerical data. To accomodate this diversity, systems designed to store, transport, display, and manage multimedia data must have considerably more functionality and capability than conventional database systems.

One of the major challenges is to synchronize various types of data in both space and time in order to compose complex multimedia objects. Here, we assume that an object-oriented model is used to represent multimedia data [Mas91, GBT94]. A multimedia object may be a combination of real-time data, such as audio and video, with the more conventional text and image data. Real-time data require time-ordered presentation to the user. Thus, a composite multimedia object may involve specific timing relationships among the different types of component media. For example, in a slide presentation, a sequence of images and speech fragments must be temporally combined and presented to compose a multimedia object. The succession of images must be kept in synchronization with that of the speech fragments. Spatial relationships, such as the relative location of image components, may also require synchronization.

The process of coordinating the location and real-time presentation of information and maintaining spatial and time-ordered relationships among component media is known as spatial and temporal synchronization [LG90]. Spatial and temporal synchronization among multimedia data will also impact on workflows when cooperative work occurs in a multimedia environment. For example, if two tasks in a workflow retrieve media objects which require synchronization, then the retrieval actions of both tasks must be executed synchronously. Thus, new dependencies that reflect those features of multimedia data must be considered in the specification of workflows in the multimedia database environment. These dependencies among tasks, termed *spatiotemporal dependencies*, pose new challenges to workflow management. Traditional theories of transaction management are inadequate to handle such synchronization requirements in workflow management.

In this paper, we discuss the effects of multimedia data on workflow specification and management in multimedia database systems. In particular, we identify two major types of the spatiotemporal dependencies, namely, progressive parallel and real-time precedence dependencies, that must be specified in workflows. These spatiotemporal dependencies demonstrate the basic steps to incorporate multimedia data into workflows. More complex spatiotemporal dependencies may also exist among tasks and can be formulated. These spatiotemporal dependencies among tasks propose new challenges to workflow management. The theory of traditional transaction management is inadequate to handle such synchronization requirements in workflow management.

This paper is organized as follows. Section 2 introduces the framework for multimedia workflows.

Section 3 discusses the synchronization problem within multimedia data. Section 4 addresses the effects of multimedia data on workflow specification and management. Concluding remarks and future research are offered in Section 5.

2 A Framework for Multimedia Workflows

This section introduces a framework for specifying multimedia workflows. Informally, a multimedia workflow consists of a set of relevant tasks among which various dependencies are specified regarding the control flow, data flow, and other conditions to which the tasks must conform in their executions. A task within a workflow is a sequence of operations performed on data items in a database. We now discuss various dependencies among the tasks of a multimedia workflow and introduce a formal model for such workflows.

The most common dependency placed upon tasks addresses questions of precedence. Precedence dependencies define the parallel and sequential executions among tasks. For example, if task t_i precedes task t_j, then t_i must be finished before t_j starts; otherwise, t_i and t_j can be executed in parallel. Thus, precedence dependencies define a partial order on the tasks.

Each task may also have alternatives which will be triggered in the event of its failure. The entire workflow thus may not be aborted, even if any of its tasks fails, as long as an alternative finishes. Such alternative relationships have been defined as preference (or contingency) dependency [BDS$^+$93, ZNBB94]. For example, if task t_i fails semantically, then task t_j is executed as an alternative. When such contingency dependencies are specified among tasks within a workflow, the workflow then contains alternative partial orders, with each partial order representing the execution of the entire workflow. Contingency dependences should also indicate the priority among alternative tasks for selection in completing the execution of the workflow.

Other constraints attached to tasks may define termination dependencies or critical task dependencies [BDS$^+$93]. For example, if t_j terminates in state st_i, then t_i must terminate in state st_j; and if t_i fails semantically, then the entire workflow must fail. By explicitly specifying termination dependencies, various termination relationships among tasks can be defined. Critical task dependencies allow us to identify those tasks within workflows which are so distinguished.

Recent research in this field has also proposed that chronological dependencies [GRL93] be used to impose real-time constraints on tasks. A chronological dependency has been defined by specifying the start time and the expected completion time of a task. In the next section, we will discuss additional dependencies that have not been identified in non-multimedia situations.

Thus, the structure of a workflow can be depicted as a set of alternative partial orders, with various dependencies defined on tasks. The above dependencies and those listed below provide the flexibility and functionality to specify workflows. A formal definition is provided.

Definition 1 (Multimedia workflow) *A multimedia workflow W is a 3-tuple (W,O,D) where*

- *W is a set of tasks*
- *O is a set of partial orders on W*
- *D is a set of predicates on W*

3 Synchronization of Multimedia Data

This section discusses the issue on synchronization of multimedia data. Synchronization has been recognized as the central problem to be addresses in the design of multimedia systems [ENK+93, LG90, Ste90, LG93]. Both spatial and temporal synchronization are addressed.

Let a media element be a highly structured aggregate of simpler objects, such as video frames, audio samples, or musical notes. We assume that an object-oriented data model is supported [Mas91, GBT94] for the representation of multimedia data. Thus, a media element is termed a "media object." We consider data streams as sequences of time-dependent media elements.

We first discuss the question of spatial synchronization of multimedia data. Spatial relationships between media objects are important in image processing [CYDA88, TPF+91]. In the data representation of a two-dimensional image, information related both to position as well as distance must be considered in order to represent the location of a media object. For example, we may say that media object o_1 is east of media object o_2 or that media object o_3 is fifteen centimeters from media object o_4. These primitive spatial relations can be classified into the following two categories, although this list is not exhaustive:

- **Position**:
  - east, west, south, north, southwest, southeast, etc.
  - over, under, next-to, etc.
  - surrounded-by, partly-surrounded-by, same-position-as, etc.
- **Distance**: near, is-distance-from, etc.

Different media objects may need to be synchronized spatially. For example, one might specify that two image fragments must be located in the same space. In addition, different data streams may also need to be synchronized spatially. For instance, a motion video and its caption must be synchronized spatially at the appropriate position in a movie. We assume that such synchronization requirements are specified in the data representation.

We shall now consider the temporal synchronization of multimedia data. Temporal synchronization may be required within a single data stream. For example, one might specify that media objects located in the first quadrant must appear before those located in the second quadrant. Moreover, temporal synchronization is important in integrated media streams. Such relations have been identified as "precedence" in [Mas91]. Different data streams may interleave in a coordinated and consistent fashion within multimedia environments. In general, temporal synchronization is required in the following media integration context [Ste90]:

- *Intermedia Relationship*: A state transition or activity in which one medium affects another medium; i.e., a certain text pattern activates a moving video sequence.
- *Media Conversion*: The information contained in one medium is "translated" into information in another medium; i.e., text-to-speech conversion.
- *Media Cooperation*: Two or more media, such as audio and video, simultaneously exchange information.

These categories of media integration dictate that synchronization must be either serial or parallel. Serial synchronization requirements determine the rate at which events must occur within a single data stream; this includes the rate at which sound information is processed or video information is fetched. Parallel synchronization requirements determine the relative scheduling of separate data streams. In this context, different continuous data streams must be semantically synchronized. For example, to be meaningful, a video stream must be properly synchronized with a voice stream. Note that, in non-trivial multimedia applications, a data stream may have both a serial synchronization requirement and a parallel relationship with other streams. Furthermore, some multimedia applications may require composite or embedded synchronization.

These aspects of synchronization can affect the partitioning of tasks in workflows, which we will specify as *spatiotemporal dependencies* and discuss in the following section.

4 Spatiotemporal Dependencies in Multimedia Workflows

Synchronization of multimedia data can impose additional dependencies on tasks, which we recognize as *spatiotemporal dependencies*. Such dependencies have not been identified in non-multimedia situations. We may identify two basic types of spatiotemporal dependencies among tasks: progressive parallel and real-time precedence dependencies. We then recognize other hybrid spatiotemporal dependencies that are generated from these two basic spatiotemporal dependencies. These dependencies specify those relationships among tasks that arise from the synchronization requirements among media objects.

The precedence dependencies within a workflow provide an approach to specifying parallel execution among tasks. Those tasks that are not ordered by precedence dependencies can be executed concurrently. However, the parallelism among tasks expressed by precedence dependencies is inadequate to the enforcement of the stronger parallelism requirement engendered by the accessing of media objects.

Let tasks t_i and t_j in a workflow retrieve media objects o_m and o_n which require parallel synchronization. Let t_i be $retrieve(o_m, record_1); retrieve(o_m, record_2)$ and t_j be $retrieve(o_n, record_1); retrieve(o_n, record_2)$. To preserve the required synchronization, the retrieval operations of t_i and t_j must be progressively synchronized with the accessing of media objects o_m and o_n. That is, $retrieve(o_m, record_1)$ and $retrieve(o_n, record_1)$ must be executed simultaneously; so must $retrieve(o_m, record_2)$ and $retrieve(o_n, record_2)$. We say that task t_i is *progressive parallel dependent* on task t_j if t_i and t_j access media objects on which parallel synchronization is required. Note that such a dependency can also be extended to more than two tasks. Progressive parallel dependencies are different from the parallelism between tasks specified by precedence relations. This dependency requires not only parallelism between tasks but also synchronized execution between operations within tasks. In the above example, t_i and t_j should be executed in parallel. Furthermore, the operations which are bound by parallel synchronization between media objects must be executed simultaneously.

Clearly, the precedence graph of a workflow also provides an approach to specifying sequential execution among tasks. However, the time-dependent characteristic of media objects requires that stronger time-dependent sequential relationships be specified among tasks. The sequence among tasks expressed by precedence relations is inadequate to express such relationships.

Let tasks t_i and t_j in a workflow retrieve media objects o_m and o_n which require sequential synchronization such that o_m immediately precedes o_n by s seconds. Let t_i be $retrieve(o_m, record_1); retrieve(o_m, record_2)$ and t_j be $retrieve(o_n, record_1); retrieve(o_n, record_2)$. To preserve this time-restricted sequential synchronization requirement, the operation $retrieve(o_n, record_1)$ of t_j must start exactly s seconds after the operation $retrieve(o_m, record_1)$ of t_i has finished; the operation $retrieve(o_n, record_2)$ of t_j must start exactly s seconds after the operation $retrieve(o_m, record_2)$ of t_i has finished. We say that task t_j is *real-time precedence dependent* on task t_i if there is an operation of t_j which must start t time period after an operation of t_i finishes. Similarly, this definition can also be extended to more than two tasks. Real-time precedence dependencies combine precedence relations and time-constraints into a single dependency relationship. Such a chronological dependency was first introduced in [GRL93] to impose real-time constraints on tasks. Real-time precedence dependencies define richer meanings than chronological dependencies. Since the parallel and sequential synchronization of media objects engenders a greater complexity of both order and time constraints, chronological dependencies are inadequate to specify such relationships. Real-time precedence dependencies are thus introduced to meet this requirement.

The progressive parallel and real-time precedence dependencies discussed above are only two examples selected from the many dependencies that are caused by multimedia data. More complicated dependencies may be necessary because of the composite or embedded parallel and sequential synchronization requirements among media objects. For example, the first part of a given task may require sequential synchronization with one task while its second part may require parallel synchronization with a second task.

The spatiotemporal dependencies defined on tasks in a workflow pose new challenges to workflow management in multimedia database systems. Clearly, conventional "read" and "write" operations and concurrency control techniques are not sufficient to handle time-dependent media data access. At the system level (which is similar to transaction management in traditional database systems), synchronization of both operation and task activities must be maintained. We must provide mechanisms that can ensure that two operations can be executed simultaneously. Furthermore, there may be durational differences between operations in two tasks that must be synchronized in parallel. Consequently, a "gap" occurs in the execution of the two operations. An interesting question is how to fill such "gaps" in order to preserve progressive parallel synchronization among the operations of two tasks. In addition, we must control the timing of operation executions to ensure that a task can start precisely at the expected time. These issues will not be addressed in this paper.

5 Conclusions and Future Research

This paper has discussed the relationships between workflow and multimedia data in the multimedia database environments. The time-dependent characteristics of multimedia data result in the necessity of synchronization among different data streams. Such synchronization requirements also impose new dependencies on the tasks within workflows. Two types of spatiotemporal dependencies, namely, progressive parallel and real-time precedence dependencies, have been identified. Note that these dependencies are not exhaustive. Other spatiotemporal dependencies may also exist. Because of the space limits, we did not address them completely. These spatiotemporal dependencies are crucial for specifying workflows in multimedia database systems that support cooperative activities.

Workflow management in multimedia database systems is complicated by these spatiotemporal dependencies. To build an effective workflow management system in the multimedia environment, we must address the issues on enforcement of the spatiotemporal dependencies at several levels. First of all, we need to explore the enforcement of synchronization at the operation level. The major issue is how to ensure that multiple operations of tasks are performed at the same time period. Traditional transaction management has not addressed such time-dependent issues. Secondly, the preservation of spatiotemporal dependencies at the task level needs to be addressed. Based on the operations synchronization, this level needs to maintain that the executions of operations in tasks follow the spatiotemporal dependencies. Finally, the spatiotemporal dependencies may also exist between tasks that belong to different workflows. Consequently, sophisticated concurrency control mechanisms of workflows need to be developed.

References

[BDS+93] Y. Breitbart, A. Deacon, H.-J. Schek, A. Sheth, and G. Weikum. Merging Application-centric and Data-centric Approaches to Support Transaction-oriented Multi-system Workflows. *SIGMOD Record*, 22(3):23–30, September 1993.

[CYDA88] S.K. Chang, C.W. Yan, Donald C. Dimitroff, and Timothy Arndt. An Intelligent Image Database System. *IEEE Transaction on Software Engineering*, 14(5):681–688, May 1988.

[ENK+93] Seong Bae Eun, Eun Suk No, Hyung Chul Kim, Hyunsoo Yoon, and Seung Ryoul Maeng. Specification of Multimedia Composition and A Visual Programming Environment. In *Proceedings of ACM Multimedia 93*, pages 167–174, Anaheim, California, August 1993.

[GBT94] Simon Gibbs, Christian Breiteneder, and Dennis Tsichritzis. Data Modeling of Time-Based Media. In *Proceedings of ACM-SIGMOD International Conference on Management of Data*, pages 91–102, Minneapolis, Minnesota, May 1994.

[GRL93] D. Georgakopoulos, M. Rusinkiewicz, and W. Litwin. Chronological Scheduling of Transactions with Temporal Dependencies. *VLDB Journal*, 3(1), January 1993.

[IEE94] Special Issue on Computer-Supported Cooperative Work. *IEEE Computer*, 27(5), May 1994.

[LG90] Thomas D.C. Little and Arif Ghafoor. Synchronization and Storage Models for Multimedia Objects. *IEEE Journal on Selected Arears in Communications*, 8(3):413–427, April 1990.

[LG93] Thomas D.C. Little and Arif Ghafoor. Interval-Based Conceptual Models for Time-Dependent Multimedia Data. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):551–563, 1993.

[Mas91] Yoshifumi Masunaga. Design Issues of OMEGA: An Object-Oriented Multimedia Database Management System. *Journal of Information Processing*, 14(1):60–74, 1991.

[Ste90] Ralf Steinmetz. Synchronization Properties in Multimedia Systems. *IEEE Journal on Selected Arears in Communications*, 8(3):401–412, April 1990.

[TPF+91] A. Turtur, F. Prampolini, M. Fantini, R. Guarda, and M.A. Imperato. IDB: An Image Database System. *IBM Journal of Research and Development*, 35(1):88–96, January 1991.

[ZNBB94] Aidong Zhang, Marian Nodine, Bharat Bhargava, and Omran Bukhres. Ensuring Relaxed Atomicity for Flexible Transactions in Multidatabase Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Minneapolis, May 1994.