

A **Taxonomy** of Adaptive Workflow Management

Yanbo Han, Amit Sheth 1

Large Scale Distributed Information Systems Lab

The University of Georgia, Athens, GA30602-7404

{yanbo, amit}@om.cs.uga.edu

Christoph Bussler

Applied Research and Technology, The Boeing Company

P.O. Box3707, m/s 7L-70, Seattle, WA98124-2207

Christoph.Bussler@pss.boeing.com

Abstract

This paper highlights the need for adaptive workflow management, classifies various types of workflow adaptation, and discusses potential mechanisms for achieving adaptive workflow management. Our current efforts on adaptive workflow research are presented with emphasis laid on human involvement. The developments are based on dynamic composition of workflow resources, including workflow models, as well as a strong association between exception handling and adaptive workflow management.

Keywords: Adaptive Workflow Management, Workflow Technology, Workflow Evolution, Ad-hoc Changes, System Re-configuration, and Exception Handling

1. Introduction

Under the banner of process orientation, the shift of emphasis from the quality movement and the development of vertical applications in the 80's to the engineering of organization's horizontal business processes in the 90's can be observed. Aiming at facilitating business processes and integrating independent, heterogeneous and often distributed information systems, workflow technology has gained much attention in recent years [GeHo95] [JaBu96].

A major concern of workflow technology is to model business process logic properly and provide suitable runtime support for its execution. Technically, business processes are implemented with workflow processes that are composed of a set of causally and temporally related workflow tasks. To describe and enforce such workflow processes, executable workflow models are built with a workflow modeling language. The essence of workflow modeling includes:

- Specifying business tasks and their interdependencies like control and data flow

- Specifying the application of workflow resources to individual tasks, such as deployment of application programs and assignment of roles

Task specifications define a set of logic units of work, their types and interfaces. Task scheduling reflects task inter-dependencies and execution sequences. Workflow resources include: 1) application programs or workflow model fragments that implement tasks, 2) machine agents or persons who are in charge of executing tasks, and 3) data required for performing tasks. Resource application to a task defines specific resources or resource types to be applied for performing the task. Depending on workflow modeling languages, the specification of resources is included in workflow models in different ways. Commonly, specifications of application data structures, application functions, and organization structures are represented in separate models. Even for describing the process aspect, multiple abstraction layers are often used to cope with the complexity. While some approaches try to specify all related aspects in one compact language at a single abstraction level [FoKu95], the majority of workflow modeling approaches builds workflow models with different abstraction layers. Some workflow models have a structure specification part and a task specification part (like in METEOR [KrSh95] [ShKo96], for example), covering the scheduling of tasks and task specifications respectively. Still others use an extra third layer for describing resources and their allocation in executing tasks [Han97]. A workflow management system (WfMS) is a software system that transforms the explicit representation of a workflow model into an internal and executable format [Buss96] and provides an operational environment for workflow model execution and the administration and monitoring of workflow model instances.

In essence, enhancing efficiency, flexibility and adaptability of organizations is among the utmost goals of applying workflow technology. However, the limitation of the contemporary workflow technology prevents it from adequately supporting evolutionary and on-the-fly changes demanded by practical situations. Upon the recognition of such problems, the newer generation of workflow systems, under the buzzword - adaptive workflow or adaptive workflow management [ElKe95] [PaMi96] [Buss98] [HaSh98], strives to make workflow systems more flexible and adaptive. This paper tries to establish a basic conceptual framework for adaptive workflow management. It is organized as follows. Section 2 discusses the deficiencies of the contemporary workflow technology and highlights the necessity of adaptive workflow systems. Section 3 analyzes and classifies various types of workflow adaptation. Some existing and potential mechanisms for adaptive workflow management, including our current research efforts, are discussed in section 4. Section 5 concludes with some open questions and discussions of future work.

2. The Need for Adaptive Workflow Management

First of all, two simplified situations are examined which can occur in healthcare workflow scenarios and demand adaptive workflow management features:

- In a hospital, inpatient treatment of a patient may follow one of many possible work procedures (called treatment plans), depending on the patient's concrete situation. A responsible physician will choose a plan on the spot. The standard plans expand constantly in response to medical advancements, changes of insurance and hospital policies, discovery of new diseases, or new treatment methods, drugs, devices, etc. What makes the situation extremely complex and dynamic is that the physician may have to create a special ad-hoc plan for a special case. Enumerating all possible plans in a workflow model is under normal circumstances an impossible undertaking.
- A physician orders laboratory tests for a patient, but cannot wait for the results in case of an emergency. Thus, he/she may start a treatment process to treat the emergent case. As soon as the test results arrive (even in the middle of the current treatment), they should trigger an action to let the physician know they are available. Note that nobody knows exactly when the results will arrive.

After receiving the results, the physician may need to modify his/her care plan immediately as a consequence of the new insights.

The contemporary workflow technology is not capable of supporting systems evolution and dynamic adaptation demanded by certain practical situations, such as the ones stated above. Several major influential factors are discussed in the subsequent paragraphs.

2.1 Evolution of Workflow Management Systems

In this subsection, we focus on two major factors that call for the evolution of workflow management systems.

A) Changing Environment

Business activities and environments, as well as many engineering branches in general, are highly dynamic and subject to constant evolution. As the business climate is increasingly dynamic and competitive worldwide, redesign and optimization of existing business processes become indispensable in most organizations in order to gain better efficiency and effectiveness in the rapidly changing environments. Between radical redesigns, business processes often have to be adjusted over and over again. In fact, instead of frequent radical redesigns, the final goal is to constantly improve the business processes by evolutionary redefinition. However, workflow systems today are often built in the same way as traditional application systems. That is, as finished products, they are targeted at definite scenarios and not designed to cope with a rapidly and dramatically changing environment.

B) Technical Advances

Software systems are confronted with evolution requirements caused by technical advances. Technical advances often lead to systems reconfiguration, due to, for example, replacement and updating of software components, addition of new components, and change in component interfaces.

2.2 Ad-hoc Derivation during Workflow Execution

Several often-encountered situations requiring runtime ad-hoc derivation >from planed workflow processes are discussed below.

A. Dynamic Refinement

In certain cases, it is impossible or impractical to derive a complete workflow model specification. Due to the unavailability of a complete specification, dynamic refinement may be needed at runtime. That is, only at runtime can certain tasks be completely and definitely specified. The same applies for interdependencies as well as resources required for task execution.

B. User Involvement

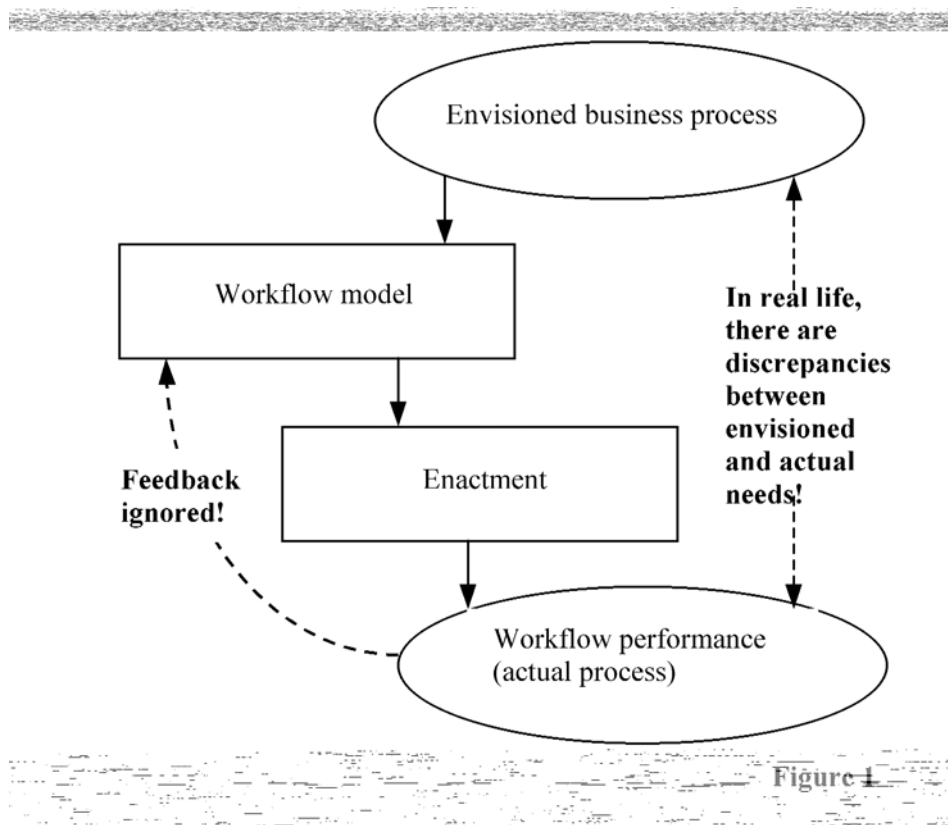
Instead of being passively controlled, certain users in a workflow system should be treated as process owners. In reality, a user has often to adjust and situate his/her activities depending on specific circumstances under the influences of external events and his/her personal working style. User's decision-making needs to be considered in executing workflow processes.

C. Unpredictable Events

As shown in the introductory examples, unpredictable events, including certain external stimuli, intervention of users, time-out, etc., should be handled properly to solve real-world problems and facilitate inter-workflow communication between different workflow processes. Furthermore, once an inter-workflow communication takes place, users or process owners must be able to respond to those events by dynamically refining the process or changing its current task and/or task interdependencies.

C. Erroneous Situations

System failure, resource conflicts and erroneous operations may cause errors and difficulties in executing workflow processes. Mechanisms for handling erroneous situations are therefore very important to a workflow system in order to ensure a smooth progress of workflow processes.



A major deficiency in most of the current workflow products and prototypes is that they assume well-structured and totally pre-definable workflow processes. Figure 1 illustrates the currently prevalent way of workflow management: Firstly, workflow models are completely built according to envisioned and reengineered versions or simply by resembling old "as-is" processes. Then, the representations (be it formal or pragmatic, graphical or textual) serve as the input for the underlying workflow management system. Finally, their execution leads to real-world processes. Note that there are, in principle, always discrepancies between envisioned processes and actual needs during workflow execution. The feedback from the real-world processes should be captured accurately and timely so that the future course of the workflow processes can be adjusted. This helps to reduce possible mismatches between models and their future real-world counterparts. To cope with the above-stated problems, contemporary workflow models must evolve from being only pre-defined plans and be open for dynamic modification and reconfiguration on the basis of the feedback information.

3. Classification of Workflow Adaptation

An appropriate taxonomy can help to find proper ways of handling workflow adaptation. We advocate the strategy of "separating concerns" for coping with adaptive workflow management. As shown in figure 2, separate layers of workflow adaptation are introduced.

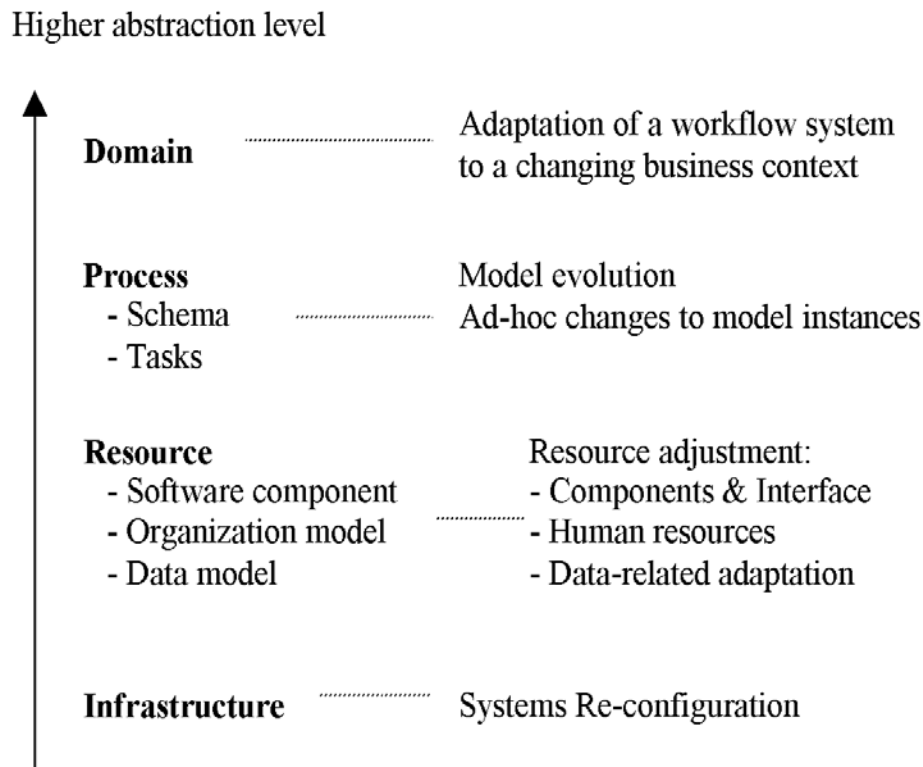


Figure 2

3.1 Adaptation in A Changing Business Context

Workflow systems do not exist for their own purposes. They are a constituent component of a business system. A business system is usually domain-specific. At domain level, a workflow system can be considered as a single component. Integrating a general-purpose WfMS in a specific business and organization context needs adaptation and reconciliation from both sides. Certain changes of the business context will also have impact on the application and embedding of a workflow system. Thus, workflow systems should be prepared to adapt themselves to a range of different business and organization settings and also to a changing context. Domain level adaptation may result in a series of adaptation requirements within a workflow system. Adaptation inside a workflow system may then take place at different levels as discussed below.

3.2 Process Level Adaptation

Process level adaptation deals with changes to workflow models and their constituent workflow tasks. It can be classified as workflow model evolution and ad-hoc modification of workflow instances. Workflow model evolution comes hand in hand with Business Process Reengineering (BPR) [HaCh93]. Thus, problems related to the redesign, substitution and version control of workflow models must be considered in accordance with business process improvements. Ad-hoc adaptation of workflow models takes place dynamically as the workflow models are being executed. As discussed earlier, this kind of adaptation is frequently needed to incorporate derivations and ad-hoc extensions in order to flexibly control workflow processes.

3.3 Resource Level Adaptation

Resource modification deals with changes and rearrangement of individual supporting workflow resources, such as substitution and modification of component interfaces of constituent software systems. Adaptation of organization and

data structures is discussed in some detail below.

3.3.1 Organization-Related Resource Adaptation

Changes in organization-related structures and resources, such as personnel changes, may have direct impact on workflow process execution. It is assumed that changes in organization structures are reflected by the corresponding changes of related workflow resources. How the changes can be promptly and effectively reflected depends on mechanisms for dealing with organization-related resource adaptation.

3.3.2 Data-Related Adaptation

Data and data structures might change during the execution of a workflow process. Usually, data that are not used by a WfMC can be changed independently and accessed by other applications at the same time. But if a workflow process depends on the existence of a piece of data or a specific property of it, then the workflow process needs to be adapted to the data changes. This of course requires that the data-related changes are made known to the workflow management system so that it can react to the changes.

3.4 Infrastructure Level Adaptation

In response to the evolution requirements and also to technical advances, the supporting software systems may need to be adapted quickly to a modified business environment or a technical setting, resulting in new system configurations. A suitable system infrastructure needs to be established so that the underlying software systems are able to efficiently support and keep up with the changes.

4. Potential Mechanisms for Adaptive Workflow Management

Suitable workflow models and flexible system infrastructures are most essential for adaptive workflow management. To develop flexible models that are open for changes, we need more flexible modeling languages. Methodologies are also needed to realize various changes and guarantee correctness. In the subsequent discussions, we mainly concentrate on workflow adaptation at the process and resource levels. Currently, concerning the adaptation of workflow models, two popular approaches are prevalent: meta-model approaches and open-point approaches.

4.1 Meta-model Approaches

Meta-model approaches utilize, explicitly or implicitly, meta-models to determine the structures and types of constituent components of workflow models. A set of primitives is usually defined with which change operations can be performed to a workflow model or a certain model instance. Examples of such an approach include ADEPTflex [ReDa97], WASA [Wesk98] and WIDE [CaCe96]. Syntactical correctness can be controlled and changes can be made by users provided that a suitable tool support is available. But attention is largely paid to structural changes of workflow models, that is, adding, deleting or skipping tasks and altering task dependencies. A general resource management strategy is not yet available so that certain conflict situations may still arise due to rearrangements of resources including human beings [Wulf96]. Local adjustments concerning a single task are often neglected. Such local adjustments are very useful for a user to situate his/her work environment, including making decision in response to a special situation and on the basis of a variety of choices, reporting exceptional cases that are out of his/her responsibility, and so on.

4.2 Open-point Approaches

Open-point approaches define special points in a workflow model, where adaptation can be made. Here, the concept of adaptation is often generalized, including provision of multiple choices for users to choose, binding of certain resources at runtime, or provision of an open interface through which the so-called "late-modeling" can be made. Late-modeling means that, during the execution of the overall model, certain sub-models can be dynamically defined and put into an immediate use. Several workflow management systems, such as MELMAC [DeGr94], MOBILE [JaBu96] and ObjectFlow [HsKl96], facilitate the open-point approach to different extents. However, it turns out to be insufficient for

dealing with unexpected situations just to have fixed, predefined open-points in models. A further development is to build a generic open interface inside all workflow tasks, as proposed in [Han97]. In this way, dynamic changes can be made more flexibly at individual tasks. A major deficiency of open-point approaches is that they have difficulties to deal with certain structural changes that are well supported in meta-model approaches.

4.3 A Synthesized Approach

We believe that the above-stated two approaches can be used complementarily for adaptive workflow management. Before we proceed to present our approach, let us examine the following two assumptions: 1) It would be beneficial to separate issues of workflow evolution and those of dynamic ad-hoc changes. More discussions will follow when we introduce the different responsibilities of process owners and process participants. 2) Traditionally, a clear separation is made between build-time and run-time in terms of workflow models. We are convinced that this is a barrier to be removed in order to make adaptive workflow feasible. Thus, in our subsequent discussion, it is implicitly assumed that changes can be made at any time.

As a redesign process, workflow evolution deals with modifying workflow models permanently to get a new version. Workflow evolution is a serious matter because it is often associated with changes of business policies and the ways of organizing business processes, not just issues of changing task sequences and ensuring correctness. A major problem is with regard to the migration issues. Certain workflow processes can have a long "life". How a running instance can be properly stopped and the valid process data, including running states, be transferred to the instances of the new model still remains a topic of research. Note that there can exist a strong association between ad-hoc changes and workflow evolution. If ad-hoc changes are to be made permanent, we are then confronted with a problem of workflow evolution.

While some changes can be performed by normal workflow users to facilitate individualization and cope with local exceptions, others many affect multiple users and therefore should be performed by system managers or process owners who are in charge of the corresponding group. The separation of different layers is very useful for allocating responsibility and controlling change rights. Workflow evolution is performed at a corresponding higher level by someone who has the special rights for making such changes. A change may effect a single model instance or all running instances. Moreover, changes may be first made to one or more model instances and then, after verification for example, are made permanent.

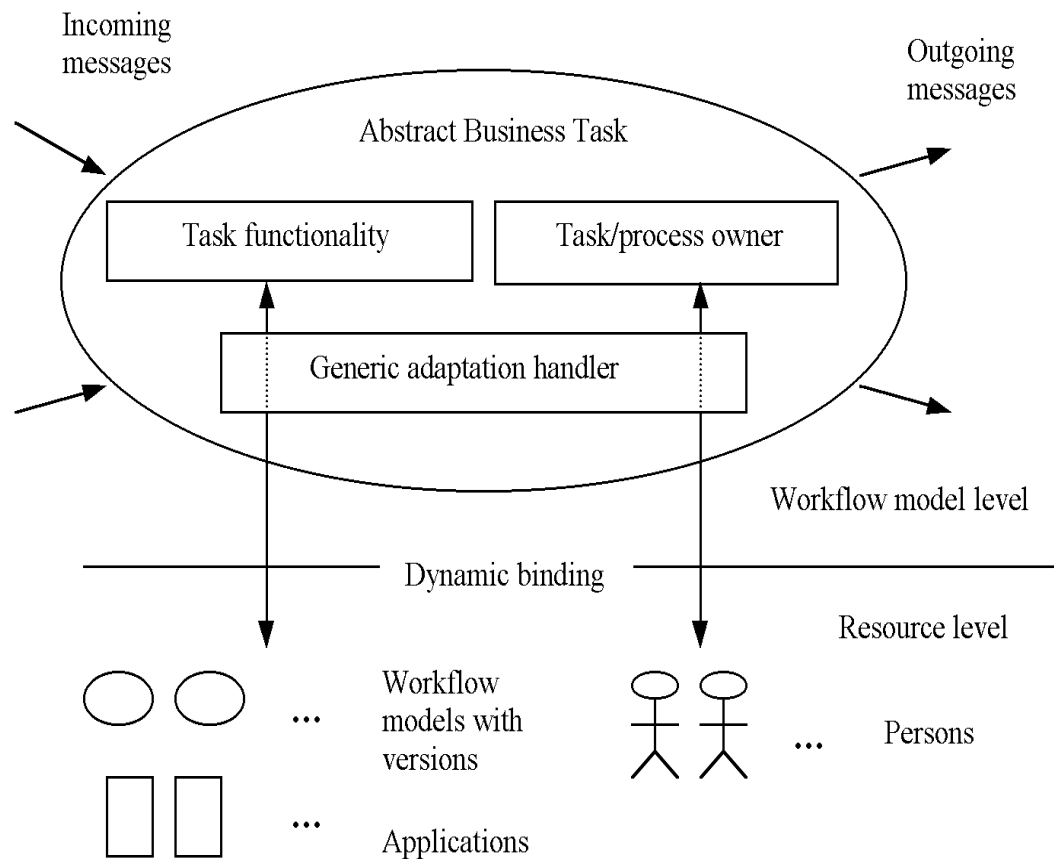


Figure 3

Our approach is sketched in Figure 3. It is based on the following observations and considerations:

A) Flexible Composition and Dynamic Hierarchy of Workflow Models

Business processes are often hierarchically organized. However, the hierarchy should not be totally fixed. This is because, while a business task can be accomplished by a workflow process at a lower level in one situation, it may be accomplished by an atomic application system or another workflow process in other situations. Generally, there can exist multiple choices for accomplishing a business task, each of which is suitable or applicable to a specific subset of all possible situations. Flexibility is significantly reduced if a business task is associated with one solution in a fixed manner. In this context, the first thing we do is to make a business task more abstract by separating its declaration from its implementation, and by allowing for a dynamic binding between a business task and various potential implementations.

B) Systematic Management and Dynamic Binding of Workflow Resources

Resource management (resource collection, allocation, assignment, recollection, etc.) forms an extra internal process. From the resource management point of view, workflow models should also be a plan of resource allocation and utilization. Tasks are executed by binding and applying various related workflow resources. Through the separation of an extra resource management layer, dynamic binding between workflow resources and workflow tasks is made feasible and safe from resource conflicts. Though most modeling languages define the application of resources to tasks in a fixed way, we advocate that resources be encapsulated in an object-oriented way, systematically managed and dynamically bound to tasks. With suitable tool support, certain changes of workflow resources can be made independently of workflow models.

C) Local Decision Making and User Involvement

Changes, especially ad-hoc changes, have to be strongly associated with user involvement. Users are either process owners who can make structural changes (cf. meta-model approaches) and process participants who can only make local

changes regarding their tasks (cf. open-point approaches). Ownership that identifies a task owner and embodies change rights is associated with business tasks. At run-time, the ownership is dynamically resolved by concrete users who become run-time task owners. Based on their ownership, users can initiate changes for modifying the deployment of all potential resources for performing their tasks, including application systems or workflow models (see Figure 3).

Structural changes may affect a number of participants. Only process owners can initiate structure changes to workflow models accessible to them, either making new versions or changing running instances of the models. Normal users are not encouraged to make structure changes to the workflow model in which they are only process participants. Since a process owner understands the logic of the workflow model as a whole, he/she knows the implication of changes in contrast to a participant who is unaware of the consequences of a change. As a participant of a workflow process, a user delegates the need of structure changes to the process owner instead of changing the flow structure by him/herself. On the other side, the process owner can delegate ownership to a user if appropriate and subsequently the user can apply structural changes. Note that a participant is the owner of the resources associated with his/her task. In particular, if the task is implemented by a workflow process, the user becomes the owner of the workflow process and can make structural changes in the corresponding workflow model to create a new version. To sum up, suppose a user is assigned a business task that is to be realized by a workflow process in which multiple users are involved. Only he/she, not one of the participating members, then has the right to make new versions of the corresponding workflow model by making structural changes.

D) Exception Handling and Adaptive Workflow Management

Mechanisms for exception handling and adaptive workflow can be tightly combined into a generic adaptation handler as shown in figure 3. Through the generic resource binding and adaptation handling mechanism embedded in each task, a user can try different variants, dynamically acquire new resources, report problems, issue commands to retry, postpone, or abort tasks, and seek help from the process owner or a system manager. Mechanisms for adaptive workflow management will also enhance the exception handling capability. For example, if certain exceptions happen too often, then it may mean that the corresponding workflow model has flaws and needs to be modified to incorporate this part into the normal flow.

E) Infrastructure Support

Besides the development of more suitable modeling formalisms, tool support is an absolute necessity. Although infrastructure issues are not treated in depth in this paper, they should not be neglected. Recent advances in, for example, distributed computing infrastructure, software architecture, agent technology, middle-ware and component based reuse methodologies, as well as methodologies for version and configuration control can all contribute to a more flexible and configurable workflow infrastructure.

5. Conclusion

So far, we have presented an initial taxonomy of adaptive workflow and our research efforts on mechanisms for adaptive workflow management. Still, there are a number of problems to be solved effectively:

- How can an adaptive workflow be analyzed prior to execution? Due to the fact that the detailed picture of an adaptive workflow process only becomes known after the process execution, adaptive workflow processes are more difficult to be analyzed than ordinary ones.
- How can an adaptive workflow process be effectively monitored?
- How can we ensure consistency and correctness related to structure changes, such as deleting, bypassing, repeating and inserting tasks? How to deal with multiple versions?
- How can we guarantee system and data integrity in a distributed environment where changes may take place simultaneously?
- Can an advanced transaction mechanism be adapted to the workflow process level?
- What are the criteria of making changes and how to control change rights more effectively?
- How can we identify and cope with side effects of changes? For example, changes of organization model, data model or

a concrete workflow resource may directly or indirectly affect workflow models, and vice versa.

1: Part of the research work is done within the METEOR project, supported partially under a cooperative agreement between National Institute of Standards and Technology's Advanced Technology Program (under the HIIT grant 70NANB5H1011) and the Healthcare Open System and Trial, Inc. consortium.

References

- [Buss96] Bussler, C.: "Specifying Enterprise Processes with Workflow Modeling Languages", In: Concurrent Engineering: Research and Applications. Special Issue: Application of Enterprise Modeling Languages for Concurrent Engineering. Vol. 4, No. 3, September 1996
- [Buss98] Bussler, C.: "Adaptation in Workflow Management", In: Proceedings of the Fifth International Conference on the Software Process (ICSP5). Computer Supported Organizational Work. Lisle, Illinois, USA, June 1998
- [CaCe96] Casati F., Ceri S., Pernici B., Pozzi G. "Workflow Evolution", Proceedings of 15th International Conference on Conceptual Modeling (ER'96), Germany, October 1996, pp.438-455
- [DeGr94] Deiters W., Gruhn V. "The FUNSOFT Net Approach to Software Process Management", International Journal on Software Engineering and Knowledge Engineering, Vol. 4, No.2, 1994
- [ElKe95] Ellis C. A., Keddara K., Rozenberg G. "Dynamic Change within Workflow Systems", Proceedings of the Conference on Organizational Computing Systems, pp.10-21, 1995
- [FoKu95] Forst A., Kuhn E., Bukhres O. "General Purpose Workflow Languages", Distributed and Parallel Databases, 3(2), pp.187-218, April 1995
- [GeHo95] Georgakopoulos D., Hornick M., Sheth A. "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure", Distributed and Parallel Databases, vol.3, pp.119-152, 1995
- [HaCh93] Hammer M., Champy J. "Reengineering the Corporation", Harper Business, 1993
- [Han97] Han Y. "Software Infrastructure for Configurable Workflow Systems - A Model-Driven Approach Based on Higher-Order Object Nets and CORBA", Wissenschaft und Technik Verlag, Berlin, 1997
- [HaSh98] Han Y., Sheth A. "On adaptive Workflow Modeling", Proceedings of 4th International Conference on Information systems, Analysis and Synthesis, Florida, July 1998
- [HsKl96] Hsu M., Kleissner C. "ObjectFlow: Towards a Process Management Infrastructure", Distributed and Parallel Databases, Vol.4 1996, pp.169-194
- [JaBu96] Jablonski S., Bussler C. "Workflow Management: Modeling concepts, Architecture and Implementation", International Thomson Computer Press, 1996
- [KrSh95] Krishnakumar N., Sheth A. "Managing Heterogeneous Multi-system Tasks to Support Enterprise-wide Operations", Distributed and Parallel Databases, 3(2), April 1995, pp.155-186
- [PaMi96] Pareschi R., Michelis G., Sarin S. (organizers) "Workshop on Adaptive Workflows" in Proceedings PAKM'96, Basel, Switzerland, 1996
- [ReDa97] Reichert M., Dadam P. "A Framework for Dynamic Changes in Workflow Management Systems",

Proceedings of 8th International Workshop on Database and Expert Systems Applications, France, September 1997, 42-48

[ShKo96] Sheth A., Kochut K. J., Miller J., Worah D., Das S., Lin C., Palaniswami D., Lynch J., Shevchenko I. "Supporting State-wide Immunization Tracking using Multi-Paradigm Workflow Technology", Technical Report TR-96-001, LSDIS Lab, Dept.of CS, Univ. of GA, February 1996.

[Wesk98] Weske M. "Flexible Modeling and Execution of Workflow Activities", Submitted to 31st Hawaii International Conference on System Sciences (HICSS-31), 1998

[Wulf96] Wulf V. "Konfliktmanagement bei Groupware", Ph.D. Thesis, University of Dortmund, 1996