Proc. of the 4th International Conference on Information Systems Analysis and Synthesis, Orlando, FL, July 1998

On Adaptive Workflow Modeling

Yanbo Han and Amit Sheth

Large Scale Distributed Information Systems Lab (LSDIS)

Computer Science Department, The University of Georgia

Athens, GA30602-7404

Abstract

As the contemporary workflow technology prevails, its incapability of supporting evolutionary and on-the-fly changes demanded by many practical situations also becomes more evident. Focusing mainly on modeling issues, the paper tries to identify related challenges, examine existing approaches and discuss potential improvements for our ongoing and future research work.

Keywords: workflow technology, workflow modeling, workflow evolution, adaptive workflow

1. Process Orientation and Workflow Technology

A business process aims at a set of defined outcomes driven by certain business goals, and is composed of a set of logically and temporally related business tasks whose fulfillment usually involves human participation. Under the banner of process orientation, the shift of emphases from the quality movement and the development of vertical applications in the 80's to the 90's engineering of organization's horizontal business processes can be observed. This is motivated by the following finding: while the automation of individual business tasks improves efficiency and productivity of single steps, business processes play a key role for achieving an improvement in competitiveness for a whole organization. Process-oriented, modern business systems are turning to goal-driven, computerized, human-operated, integrated and cooperative systems in which human participants work and interact with the aid of computer software to reach intended goals. As a technical enabler for such business systems, and also seen as a means to integrate application systems and human participants, workflow technology becomes one of the most significant and propitious metaphors in the information processing field. The key idea of workflow technology is to abstract and separate process logic from concrete application systems and use the process logic for explicit control and coordination.

Emphasizing the operational aspects of a business process, a workflow process deals with a coordinated execution of interconnected business tasks by human participants and/or computer software. To describe such workflow processes, models are built. Since multiple facets and perspectives, such as organization, process, usage, data and application functions, are involved, a single grand model may turn out to be less useful than models focusing on salient features of reality from different perspectives [MiSh97-1]. In this paper, workflow models (also called workflow specifications, workflow definitions, workflow schemata, or workflow maps in literature) refer to executable process models from the workflow process perspective. Workflow modeling refers to the activities that capture a workflow process in the form of a workflow model. A workflow model is built with a given medium (often defined by a language) and serves as one sort of inputs for a workflow management system that directly or indirectly supports the given medium. Being a coherent set of software tools, a workflow management system (WFMS or workflow software, as often called) provides an operational environment for workflow modeling as well as the execution, administration and monitoring of modeled workflow processes. Execution of workflow models will then result in real workflow processes.

In recent years, many workflow software products have been developed and marketed by the software industry (Action Workflow, COSA, FileNet, FlowMark, InConcert and SAP Business Workflow, just to name a few). Meanwhile, a number of research projects also concentrate on the workflow scenario or certain parts of it, aiming mainly at searching for a suitable methodological support. Some examples of the research prototypes include: CORMAN [DeGr94], WAM [FaWi96], ADEPTflex [ReDa97], CRYSTAL [McLe97], INCOME/STAR [ObSc94], INCOME/WF [ObSc97], MENTOR [WoWe96], METEOR[1] [KrSh95] [ShKo96] [MiSh97-2], MOBILE [Jabl94], ObjectFlow [HsKl96], Panta Rhei [EdGr97], WASA [Wesk98], and WIDE [CaCe96]. Efforts on standardizing workflow technology are also being made by the industrial coalition bodies WfMC [WfMC] and OMG [OMG], and by initiatives like SWAP [SWAP].

2. Workflow Modeling

2.1 Essence of Workflow Models

Workflow modeling is a process of abstraction, synthesis, and transformation, extracting most relevant and important properties of business processes into workflow models and oppressing less important ones. The importance is always a relative factor in a defined context. Therefore, there can be many different workflow models for the same business process, depending on the viewpoints of modelers and the

[1] METEOR is also available now as a commercial product from Infocosm Inc.

modeling mediums used. However, the essence of workflow modeling remains more or less the same:

- Specifying business tasks and their scheduling
  Task specifications define a set of logic units of work, their types and interfaces. Task scheduling reflects task inter-dependencies and sequences, normally in the form of conditional precedence relations.
- Specifying the application of workflow resources to individual tasks
  Workflow resources (or resources, for short) include application programs or workflow model fragments that implement tasks, persons or machine agents that are in charge of performing tasks, and application data as well as process data needed for performing tasks. Resources are normally further classified into various resource types. Here, we would like to devote a special attention to process data since they can play an important role in enhancing the flexibility and adaptability of workflow models. Process data can be used to complementarily describe workflow processes. They may define extra attributes and rules used in describing workflow process constraints, details of workflow process components, as well as the binding of other types of resources to tasks. For example, process data may include constraints like time stamps, data reflecting running status, references to external applications, etc. What can be categorized and treated as process data and how they are associated with tasks are usually dependent on modeling languages.
  Resource application to a task defines concrete resources or resource types to be applied for performing the task. Most modeling mediums define the application of resources to tasks in a fixed way. Only a few of them allow for a dynamic binding of resources and support a run-time rearrangement and management of resources (see [Han97-1] for a brief summary).
  Also, depending on modeling languages, the specification of resources is included in or associated with 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. While some people try to specify all related aspects in one compact language at a single abstraction level [FoKu95], the majority builds workflow models with different abstraction layers. Some workflow models have a structure specification part and a task specification part (like in METEOR, 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 [Han97-2].
  Modeling languages will also play the most essential role in deriving appropriate models for adaptive workflow. Thus, in the subsequent subsection, we first give a brief overview of the most prevalent modeling paradigms.

2.2 Prevalent Modeling Paradigms

Various modeling languages, new or established, have been used in commercial systems and research prototypes for workflow modeling. Often, they only cover certain parts of the overall workflow modeling capabilities, with other parts relying on more ad-hoc approaches. Different languages and their associated approaches form various paradigms. The major paradigms can be categorized into:

- Procedural workflow languages
  Here, efforts involve extending high-level imperative programming languages for workflow modeling. For example, the procedural workflow languages C&CO [FoKu95] and APPL/A [Oste87] are high-level imperative programming languages based on C and Ada respectively.
- Workflow scripts
  Scripts are composed of specialized, textual and descriptive primitives. For example, MOBILE uses a script language for workflow modeling.
- Transactional workflow
  Under the term transactional workflow, some people, but not all[2], view workflow management as a generalization of transaction management. In this connection, workflow models are based on ATM (Advanced Transaction Models) [AlAg96] [GeHo94].
- ECA rules
  TriGSflow [KaPr95], MARVEL [Kais89] and some other research projects [Junk95] [PeSc92] have used Event-Condition-Action (ECA) rules for workflow modeling[3].
- Active database
  Active database approaches are extended rule-based approaches with the support of active database technology. Workflow is specified with rules and workflow execution is based on trigger mechanisms provided by an active database [EdGr96] [Joos94].
- Goal directed and/or agent based
  Emphasizing local decision making and autonomy, such approaches have been used in [JoKl97] [LoWa95] [Yu96]. Similar to rule-based approaches, such approaches often lack a clear overall picture of workflow processes.
- Communication based
  In such approaches, a workflow process is reduced to communication loops between customers and performers. As an example, the Business Design Language in the Action Workflow System [MeWi92] developed by Action Technology uses speech-act based language/action perspectives to build workflow models.
- Mathematical models
  Examples of mathematical approaches to process modeling include, for example, CCS (Communicating Concurrent Systems) [Miln89], CSP (Communicating Sequential Processes) [Hoar85] and CTL (Computation Tree Logic) [AtSi96]. However, these approaches have not been applied pervasively for workflow modeling mainly because they are not (visually) intuitive and directly executable, and lack a proper means to model human involvement and to enable advanced abstraction mechanisms.
- State-transition diagrams
  In this category, workflow models are diagrams based on state-transition concepts. Petri nets [Pete81] and Statecharts [Hare87] are the most prominent ones. The Mentor project incorporates Statecharts as its workflow modeling formalism [WoWe96]. A large number of the prevalent approaches are based on Petri nets, including various extensions of Petri nets (see e.g. [BaFu93] [DeGr94]

[2] For example, the authors of [WoSh97] have argued against it.
[3] Some of these projects actually focused on software process management and automation.

[HaHi96]). Other variations, such as the Activity Networks and the Event-driven Process Chain [Sche95], have been used in FlowMark and SAP Business Workflow .

- Informal and pragmatic diagrams
  In many contemporary commercial products, informal diagrams and pragmatic icons are used to represent tasks and the routing of both tasks and documents between persons. Such icons are derived from application-specific domains or vertical applications (e.g. email routing, imaging, document flow, etc.). They often lack formal semantics and are strongly associated with the corresponding specific workflow software.

Note that the above categorization is not mutually exclusive. For example, event-condition-action rules can be found in many of the above-stated paradigms; state-transition paradigms are also considered as mathematical models, since they have strong mathematical background and have formal, rigorous semantics. While some of them derive purely textual descriptions like the procedural language paradigm and mathematical models, most of these paradigms seek help from graphical representations, because graphical representations are generally more intuitive and easier to be understood. Actually, approaches that combine some of the above-stated paradigms turn out to be more powerful and versatile [HeKr96]. Also note that, since models established with a formal formalism can provide an adequate basis for model analysis and correctness reasoning, formal foundation and methods become increasingly attractive to workflow modeling.

3. Adaptive Workflow - What and Why

In essence, enhancing flexibility, adaptability and efficiency of business organizations is one of 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. One main reason is that workflow products and prototypes of the current generation assume well-structured and totally pre-definable business 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 only. Then, they serve as the main inputs 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. 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 chances of mismatches between models and their 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. On the other side, 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.

The newer generation of workflow systems, under the buzzword - adaptive workflow [PaMi96], strives to cover both structured processes to ensure correctness and discipline as well as ad-hoc processes to increase flexibility. Upon the recognition of the dynamic changes and evolution problems, adaptive workflow becomes a recent research topic that shows a great deal of research and practice potentials. In the context of adaptive workflow, we identify the following four kinds of adaptation:

- Workflow evolution
- Ad-hoc adaptation
- Resource modification

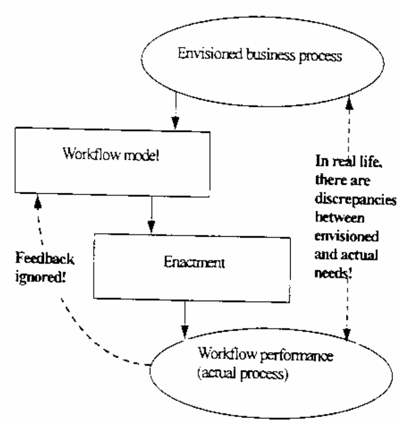

Figure 1

- System configuration

Workflow evolution comes hand in hand with Business Process Reengineering (BPR) [HaCh93]. Business activities and environments, as well as many engineering branches in general, are highly dynamic and subject to constant evolution. Globally, as the business climate is increasingly dynamic and competitive worldwide, the ability to be responsive to rapid changes is an essential success factor for enterprises. Thus, in most organizations, redesign and optimization of existing business processes become indispensable in order to gain better efficiency and effectiveness in the rapidly changing environments. Apparently, it is not enough to radically redesign the business processes once and for all. Between radical redesigns, business processes often have to be adjusted over and over again. 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 usually takes place dynamically as the workflow models are being executed. This kind of adaptation is frequently needed to incorporate derivations and ad-hoc extensions in order to flexibly control workflow processes. For example, at the very fundamental level, a clerical worker has often to adjust and situate his/her activities depending on concrete circumstances under the influences of external events and his/her personal working style. Furthermore, dynamic adjustments may be needed due to the unavailability of a full specification at the beginning of the execution or the ad-hoc handling of unexpected situations, such as external intervention (intervention of other users, time-out, etc.), system failure, and infrastructure changes. Now, let us examine two simplified situations (found in a healthcare workflow scenario) that demand dynamic adaptive features:

In a hospital, a physician orders laboratory tests for a patient, but cannot wait for the results in case of emergency. Thus, he/she may start a treatment process to treat the emergent case. But if the test results arrive in the middle of the 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.

An 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 expands 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 worse 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 normally inappropriate.

Resource modification deals with changes, adaptations and rearrangement of individual supporting workflow resources, such as substitution of constituent software systems or personnel changes caused by changes in organization structures[4].

In response to the evolution requirements and also to technical advances, the supporting software systems may need to be adapted quickly to the modified business environment, resulting in new system configurations.

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, no such distinction is made and it is implicitly assumed that changes can be made at any time. Since adaptive changes may have certain ripple effects, it is helpful to classify different changes and identify possible change effects. We advocate the strategy of "separating concerns" or "divide-and-conquer". Consequently, separate layers for model modifications are introduced in our approach. We differentiate changes at four different levels: structure level, task level, resource level and system level. Structure level changes affect the interdependencies and sequences of tasks, task level changes are concerned with modifications of individual tasks, resource level changes are concerned with changes of workflow resources, and system level changes refer to adjustments of a concrete execution environment.

[4] It is assumed that changes in organizational structures or data models are reflected by corresponding changes of related workflow resources.

Evolutionary changes deal with long-term, permanent changes that are often related to the changes of organization policies. They are performed at a corresponding higher level by someone who has the special right for making such changes. Workflow evolution often results in structural changes to workflow models and possibly changes at other levels, too. In contrast, adjustments for situating personal work do not affect structure level in principle and may directly be made by the corresponding user. 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 workflow owners who are in charge of the corresponding group. The separation of different layers is very useful for allocating responsibility and controlling change right. Concerning the scope of changes, changes can be made to a workflow model or workflow model instances. 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.

A number of problems have to be solved to ensure a meaningful and correct change, however. They are related to questions like:

- Who can make what kind of changes?
- 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.
- How can we ensure the consistency and correctness related to structure changes, such as deleting, bypassing, repeating and inserting tasks?
- How can we guarantee system and data integrity in a distributed environment where changes may take place simultaneously?
- What are the criteria and limitations of changes and how to deal with multiple versions?
- Can failure and exception handling mechanisms be generically integrated with adaptive workflow?
- Can an advanced transaction mechanism be adapted to the workflow process level?
- How can an adaptive workflow be effectively monitored?
- 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.

4. Prime Time of Adaptive Workflow

4.1 Adaptive Workflow Modeling

Suitable workflow models and system infrastructure are most essential to adaptive workflow. The popular paradigms of workflow modeling introduced in section 2 fail to cope with the challenges of adaptive workflow satisfactorily. 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. Currently, concerning the adaptation of workflow models, two popular approaches are prevalent: meta-model approach and open-point approach.

Meta-model approaches utilize 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 even a certain model instance. Examples of such an approach include ADEPTflex, WASA and WIDE. 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, making decision on the basis of a variety of choices, reporting exceptional cases that are out of his/her responsibility, and so on.

Open-point approaches set up 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, MOBILE and ObjectFlow, 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 every workflow tasks, as proposed in [Han97-2]. 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.

We believe that the above-stated two approaches can be used complementarily for adaptive workflow. However, it would be beneficial to separate issues of workflow evolution and those of dynamic ad-hoc changes. As a redesign process, workflow evolution deals with modifying workflow models permanently to get a new version. Workflow evolution is a serious process 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. Concerning process reengineering, workflow evolution does not necessarily impose special tool support for making changes. Workflow models can be redesigned using ordinary editing tools as if they were new and used the existing models as referential "templates". But, 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 and running states be transferred to the instances of the new model still remains a topic for 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. In the subsequent paragraphs, we mainly concentrate on our strategies for adapting to ad-hoc changes and resource modifications. As a matter of fact, with the resource management and dynamic binding concepts, we are trying to reduce certain structural changes to quantitative resource changes. In addition, we want to make use of the potentials of model composition, resource management, and flexible binding of resources to cope with the change problems.

Our approach is sketched in Figure 2. User involvement is emphasized. 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). Based on their ownership, users can initiate changes for modifying all potential resources for performing their tasks, including application systems or workflow models as shown in Figure 2. Structural changes at the same level are not, and in our opinion should not be, supported. That is, only process owners can initiate structure changes to workflow models accessible to them, either making new versions or changing running instances of the models. As a participant of a workflow process, a user delegates the needs of changes to the process owner (instead of changing the flow structure by him/herself). 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 models to create a new version. Each task is equipped with an open, generic mechanism for binding resources and handling exceptional situations. Our approach is based on the following observations and considerations:

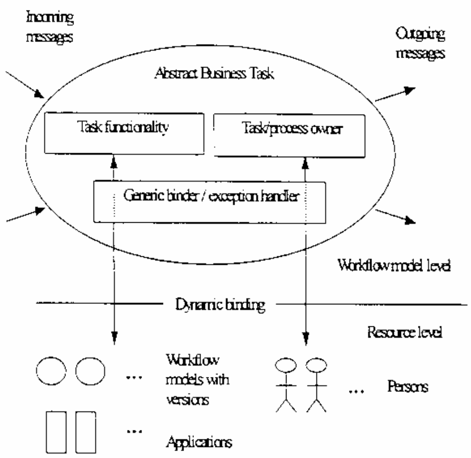

Figure 2

- 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 connection, 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.

- 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 tasks is made feasible and safe from resource conflicts. We advocate that resources be encapsulated in an object-oriented way. With suitable tool support, certain changes of workflow resources can be made independently of workflow models.

- Local decision making and user involvement

Changes, especially ad-hoc changes, have to be strongly associated with user involvement. Each responsible user should be provided with all available resources that are dynamically managed and adjusted by the underlying resource management facilities. As such, he/she is provided with a free room to make local decisions and does not need to care about resource conflicts. Through the generic resource binding and exception handling mechanism embedded in each task, a user can also dynamically acquire new resources, report problems, or seek help from a system manager or the process owner. Ownership that identifies a task owner is associated with business tasks in the form of process data. The rights of making change are embodied by such an association. At run-time, the ownership is dynamically resolved by concrete users who become run-time task owners. Moreover, if a task is implemented by a workflow process, the corresponding user is also the process owner. 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.

4.2 Multi-Paradigm Approaches

In the past several years, efforts have been made in various research projects to cope with the new challenges of adaptive workflow with multi-paradigm approaches. For example, the integration of coordination, collaboration and information management is proposed in [ShKo97] and [Shet97]. Figure 3 illustrates the basic ideas of WCCS (Work Coordination and Collaboration System). We argue that coordination technologies that are highlighted by workflow management, collaboration technologies that are highlighted by groupware, desktop conferencing and CSCW, and information management technologies that are highlighted by database and management information system should be integrated together to establish a unified framework to cope with the dynamic nature of organizations and to support a more comprehensive human participation.

Similar ideas have been pursued in the project WAM [FaWi96]. In WAM, communication-based (speech-act models) and coordination-based models (Petri net models) are used together to support inter-personal collaboration as well as asynchronous and synchronous distributed coordination of business processes in the public administration domain. Workflow processes allocate tasks and multimedia documents to staff members according to certain timing relations. A task is a meaningful work unit which can be atomic or made up of a set of ordered sub-tasks. Each task is assigned to a so-called

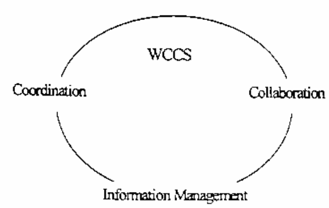

Figure 3

executor or a group of executors who are responsible for the execution of the task. All necessary information for carrying out a task, including office documents in electronic form or references to other forms of documents such as paper documents, access control, and a catalogue of available software tools, is gathered together in a folder that is associated with the corresponding task. Roles are attributes assigned to tasks, and they are linked to one or several positions in an organizational structure. At run-time, roles are replaced by responsible persons. For the tasks that are not atomic, pieces of sub-workflow related to them are modeled with Petri nets. A task is activated either through communication-based user interaction or by a net in which other nets can be called. A user can flexibly choose any possible way to fulfill a task or collaborate with others using teleconferencing to solve a complicated problem. Thus, the whole system is very flexible for supporting high-level business processes. However, there is no overall picture of workflow processes. This may be a big deficiency because workflow models are also used for documentation or for monitoring purposes. Dynamic arrangements of workflow resources are not supported either. Nevertheless, WAM, WCCS and other multi-paradigm approaches open up a new dimension for a synergic solution to flexible workflow management that is in nature a multi-disciplinary problem.

5. Conclusion

In this paper, we discussed issues associated with workflow evolution and ad-hoc adaptation. The following mechanisms or ideas are advocated with emphases:

- Abstraction of business tasks and separation between goals and implementations
- Encapsulation of implementations as workflow resources and dynamic binding of resources with tasks
- Embedding of an explicit resource management mechanism
- Emphases on user-oriented adaptive changes
- Application of multiple paradigms and integration of collaboration tools

We realized that, 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. Currently, we are extending both the modeling formalism - METEOR and its underlying WFMS with the proposed ideas. There is still a long way to go to establish an adequate methodology for adaptive workflow. The best results of the paper would be to inspire constructive ideas and fruitful discussions.

Acknowledgements

The major part of this research work is done within the METEOR project. The authors would like to thank the METEOR team, especially Prof. Krys Kochut and Prof. John Miller, for the insightful discussions.

References

[AlAg96] Alonso G., Agrawal D., et. al. "Advanced Transaction Models in Workflow Context", Proceedings of 12th International Conference on Data Engineering, 1996

[AtSi96] Attie P. C., Singh M. P., Emerson E. A., Sheth A., Rusinkiewicz M. "Scheduling Workflows by Enforcing Intertask Dependencies", Distributed Systems Engineering, Vol.3, No.4, December 1996, pp.222-238

[BaFu93] Bandinelli S.C., Fuggetta A, Ghezzi C. "Software Process Model Evolution in the SPADE Environment", IEEE Tran. on Software Engineering, Vol.19, No.12, December 1993

[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

[EdGr96] Eder J., Groiss H., Liebhart W. "Workflow Management and Databases", 2nd International Forum d'Informatique Appliquee, Tunis, March 1996

[EdGr97] Eder J., Groiss H., Liebhart W. " The Workflow Management System Panta Rhei ", NATO Advanced Study Institute's Workshop on Advances in Workflow Management Systems and Interoperability, Turkey, August 1997

[FaWi96] Faustmann G., Wikarski D. "Exception Handling in Petri-Net-based Workflow Management", Proceedings of PAKM'96, International Workshop on Adaptive Workflow, Basel, Switzerland, October 1996

[FoKu95] Forst A., Kuhn E., Bukhres O. "General Purpose Workflow Languages", Distributed and Parallel Databases, 3(2), pp.187-218, April 1995

[GeHo94] Georgakopoulos D., Hornick M., Krychniak P., Manola F. "Specification and Management of Extended Transactions in a Programmable Transaction Environment", International Conference on Data Engineering, Houston, February 1994, pp.462-473

[HaCh93] Hammer M., Champy J. "Reengineering the Corporation", Harper Business, 1993

[HaHi96] Han Y., Himmighöfer J., Schaaf T., Wikarski D. "Management of Workflow Resources to Support Run-time Adaptability and System Evolution", Proceedings of PAKM'96, International Workshop on Adaptive Workflow, Basel, Switzerland, October 1996

[Han97-1] Han Y. "HOON - A Formalism Supporting Adaptive Workflow", Technical Report #UGA-CS-TR-97-005, Department of Computer Science, The University of Georgia, November 1997

[Han97-2] 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

Hare87] Harel D. "Statecharts: A Visual Formalism for Complex Systems", Sci. Computer Program., vol.8, 1987, pp.231-274

HeKr96] Heimann P., Krapp C. A., Westfechtel B. "DYNAMITE: Dynamic Task Nets for Software Process Management", Proceedings of the 18th International Conference on Software Engineering, Berlin, 1996, pp.331-341

[Hoar85] Hoare C. A. R. "Communicating Sequential Processes", Prentice Hall, 1985

[HsKl96] Hsu M., Kleissner C. "ObjectFlow: Towards a Process Management Infrastructure", Distributed and Parallel Databases, Vol.4 1996, pp.169-194

[Jabl94] Jablonski S. "MOBILE: A Modular Workflow Model and Architecture", Proceedings of International Conference on Dynamic Modeling and Information Systems, Netherland, 1994

[JoKl97] Joeris G., Klauck C., Herzog O. "Dynamical and Distributed Process Management based on Agent Technology", 6th Scandinavian Conference on Artificial Intelligence (SCAI'97), August 1997, Finland, 1997, p p. 187-198

[Joos94] Joosten S. "Trigger Modeling for Workflow Analysis", Proceedings of CON'94 - Workflow Management, Challenges, Paradigms and Products, 1994, pp.236-247

[Junk95] Junkemann G. "A Dedicated Process Language Based on EER-models, Statecharts and Tables", Proceedings of 7th International Conference on Software Engineering Environment, 1995

[Kais89] Kaiser G. E. "Rule-Based Modeling of the Software Development Process", ACM SIGSOFT Software Engineering Notes, 14(4), 1989, pp.84-86

[KaPr95] Kappel G., Pröll B., Rausch-Schott S., Retschitzegger W. "TriGSflow - Active Object-Oriented Workflow Management", Proceedings of the 28th Hawaii International Conference on System Sciences (HICSS '95), January 1995, pp. 727 - 736

[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

[LoWa95] Lockemann P. C. and Walter H. D. "Object-Oriented Protocol Hierarchies for Distributed Workflow Systems", Internal Report No.42, University of Karlsruhe, 1995

[McLe97] McClatchey R., LeGoff J., Baker N., Harris W., Kovacs A. "A Distributed Workflow and Product Data Management Application for the Construction of Large Scale Scientific Applications", NATO Advanced Study Institute's Workshop on Advances in Workflow Management Systems and Interoperability, Turkey, August 1997

[MeMo95] Merz M., Moldt D., Müller K., Lamersdorf W. "Workflow Modeling and Execution with Coloured Petri Nets in COSM", Workshop on Applications of Petri Nets to Protocols, Proceedings of 16th International Conference on Application and Theory of Petri Nets, 1995

[MeWi92] Medina-Mora R., Winograd T., Flores R., Flores F. "The Action Workflow Approach to Workflow Management Technology", Proceedings of the 4th Conference on CSCW, October 1992, pp.281-288

[Miln89] Milner R. "Communication and Concurrency", Prentice Hall, 1989

[MiSh97-1] Miller J., Sheth A., Kochut K., "Perspectives in Modeling: Simulation, Database and Workflow", ER97, October 1997

[MiSh97-2] Miller J., Sheth A., Kochut K., Palaniswami D., "The Future of Web-Based Workflows", Proc. of the International Workshop on Research Directions in Process Technology, Nancy, France, July 1997

[ObSc94] Oberweis A., Scherrer G., Stucky W. "INCOME/STAR: Methodology and Tools for the Development of Distributed Information Systems", Information Systems, Vol.19, No.8, 1994, pp.643-660

[ObSc97] Oberweis A., Schaetzle R., Stucky W., Weitz W., Zimmermann G. "INCOME/WF: A Petri Net Based Approach to Workflow Management", Proceedings of Wirtschaftsinformatik'97, Berlin, Feburar 1997

[OMG] http://www.omg.org

[Oste87] Osterweil L. "Software Processes Are Software Too", Proceedings of the 9th International Conference on Software Engineering (ICSE), March 1987, pp.2-13

[PaMi96] Pareschi R., Michelis G., Sarin S. (organizers) "Workshop on Adaptive Workflows" in Proceedings PAKM'96, Basel, Switzerland, 1996

[PeSc92] Peuschel B., Schäfer W. "Concepts and Implementation of a Rule-based Process Engine", Proceedings of 14th International Conference on Software Engineering, 1992, pp.262-279

[Pete81] Peterson J. L. "Petri Net Theory and the Modeling of Systems", Prentice-Hall, 1981

[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

[Sche95] Scheer A. W. "Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse", Springer, 1995

[Shet97] Sheth A. "From Contemporary Workflow Process Automation to Adaptive and Dynamic Work Activity Coordination and Collaboration", Siggroup Bulletin, Vol.18, No.3, December 1997, pp.17-20

[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.

[ShKo97] Sheth A., Kochut K. J. "Workflow Applications to Research Agenda: Scalable and Dynamic Work Coordination and Collaboration Systems", NATO Advanced Study Institute's Workshop on Advances in Workflow Management Systems and Interoperability, Turkey, August 1997, pp.34-59

[SWAP] http://people.netscape.com/kswenson/SWAP/

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

[WfMC] http://www.aiim.org/wfmc

[WoSh97] Worah D., Sheth A. "Transactions in Transactional Workflows", in Advanced Transaction Models and Architectures (S.Jajodia and L. Kerschberg, Eds.), Kluwer Academic Publishers, 1997

[WoWe96] Wodtke D., Weissenfels G., et. al. "The Mentor Project: Steps towards Enterprise-wide Workflow Management", Workshop Proceedings of Workflow and Process Automation in Information Systems, Greece, 1996
[Wulf96] Wulf V. "Konfliktmanagement bei Groupware", Ph.D. Thesis, University of Dortmund, 1996
[Yu96] Yu L. "A Coordination-based Approach for Modeling Office Workflow", International Symposium on Business Process Modeling, Germany, October, 1996