AN OPEN ARCHITECTURE FOR ADAPTIVE WORKFLOW MANAGEMENT SYSTEMS

R. Siebert
University of Stuttgart
Institute of Parallel and Distributed High-Performance Systems (IPVR)
Stuttgart, Germany

One of the main tasks of workflow management systems is the coordination of activities in business processes. This paper introduces a solution framework to support adaptive workflows with complex and unstructured task dependencies.

In the first section, the research direction is outlined and a short overview of the PoliFlow project is given. Then, requirements for enterprise process management systems are identified with a special focus on coordination and control of unstructured processes.

Taking these requirements as a basis, an integrated approach for adaptive workflow support is presented. A flexible workflow model including advanced control structures allows actors to modify workflow instances during runtime. Finally, a reference architecture is proposed. On the basis of the SWATS system, it is illustrated how an adaptive support layer can be integrated to extend existing workflow management systems. The paper concludes with an overview of the current state of the work and future issues.

1. Introduction

1. 1. Workflow Systems and Adaptability

The evolution of workflow management systems (WFMS) has been influenced by several application types and application areas like insurance or banking. Traditionally, WFMS have been applied to support task coordination in well defined and frequently used processes to achieve quantitative benefits like performance improvements or cost reductions. During the past few years, qualitative potentials of process management have been realized. Enormous effort has been made in improving and installing of WFMS.

Although advantages of WFMS are well known, the number of successful workflow projects is still small. One of the reasons for this is that WFMS represent a very strategic form of information technology which requires a high level of integration with other information systems. Return of investment highly depends on the extent to which the systems can be applied. Thus, different kinds of processes have to be supported by WFMS (Wolf/Reimers, 1996).

In the literature, different types of processes or workflows are distinguished (e.g. production, scientific, ad hoc etc.) (Wainer et al., 1996). Most workflow or groupware systems focus on one special process type while others are supported insufficiently. To realize the idea of enterprise process management, different types of processes with various levels of structuring must be executable. One possible approach is the extension of traditional workflow concepts and architectures to control semi-structured or unstructured processes, too.

1. 2. Terms and Definitions

Some terms like *adaptive workflow*, *structured* and *unstructured processes* or *ad hoc workflows* are frequently used in the literature but with varying meanings. Especially, the terms 'structured workflow' and 'unstructured workflow' have only been described implicitly by giving examples. Nevertheless, these terms are frequently used to classify processes by outlining special skills or lack of existing systems.

Describing the structure of a finished workflow seems to be easy, but performing a workflow means instantiation and execution of a predefined process definition. Therefore, performing processes in the desired way requires every detail to be well known during buildtime (Wainer et al., 1996). Examples of ad hoc processes described in the literature all have in common that process relevant information is either difficult to determine or can only be determined during runtime (Sheth, 1997).

From this point of view, the structure of a process has to be estimated in context of a special organizational and technical environment, or in context of a virtual environment, as offered by a typical WFMS. It mainly depends on:

- the time when information will be available (buildtime, instantiation, runtime) and
- the way or circumstances how information can be gathered (e.g. given by definition, automatically extractable, interactive/manual).

If there are relevant details for workflow execution that cannot be fixed in the workflow definition, there is a need for workflow enactment which enables various ways of workflow instance execution. In that sense adaptability in the context of workflow management means that unstructured processes can be performed. The ability of a system to perform unstructured workflows depends on the expressiveness of the workflow model and the power of the workflow enactment.

1. 3. Related Work

Looking at the evolution of workflow technology with respect to adaptability two phases can be made out. First, structured and unstructured work has been strictly separated. Then it has been recognized that this distinction is insufficient because most processes include both structured and unstructured parts. Moreover, there is a seamless transition between them (Wolf/Reimers, 1996).

Early approaches tried to combine structured and unstructured work by systems integration. On the one hand, workflow functionality was added to groupware platforms like Lotus Notes, e.g. to achieve ad hoc routing. On the other hand, unstructured tasks have been added to workflow systems using cooperative activities or synchronous telecooperation services have been integrated with workflow support (Siebert et al., 1997).

Thereafter, several approaches were chosen to improve flexibility of task coordination by adding complex control structures (Jablonski, 1994), dynamic versioning mechanisms (Amberg, 1996), descriptive modeling (Jablonski et al., 1997) or modifications of workflow scripts (Reichert/Dadam, 1997; Weske, 1998). Some of these concepts can be found in particular commercial systems (McCarty/Sarin, 1997) or certain integrated solutions combining several approaches (Siebert, 1996).

The solution approach presented in this paper should be applicable for various workflow management systems. On the one hand automatic and manual adaptations have been explicitly combined and on the other hand advanced control of manual adaptations has been added.

1. 4. The PoliFlow Project

The PoliFlow research project was funded by the German Secretary for Education, Science, Research, and Technology (BMBF). PoliFlow had the mission to develop a process management solution to support business processes in the German public administration. Due to the variety of process types in this application area, there is a need for highly flexible workflow functionality to support inter-organizational workflows containing structured and unstructured process .
For that reason the scientific work in PoliFlow concentrates on two major research directions, flexibility and adaptability of workflow execution and security issues in inter-organizational workflows (Kindler, 1997). To meet the requirements of the widely distributed and heterogeneous application area, asynchronous workflow services and synchronous telecooperation services had to be integrated and Internet technology had to be considered parts (Siebert et al., 1997). Based on new and demanding technologies like Java and CORBA, Stuttgart's Workflow And Telecooperation System (SWATS) has been developed. This system will be presented in the following. SWATS is used in PoliFlow as a prototypical process management solution as well as an open platform where latest results from our research areas could be integrated.

2. Requirments for Enterprise Process Management

In this section, the public administration is shortly described as a typical scenario where most requirements have been analyzed. Then, relevant requirements are listed and discussed.

2. 1. Public Administration - an Appropriate Scenario

Public administration represents a huge enterprise dealing with all kinds of tasks (Siebert, 1996). Various characteristics of the user environment represent the challenges of an advanced enterprise workflow management system. In spite of a certain autonomy of each authority, there are lots of connections between them. As a result they need coordinative and cooperative support for a variety of business processes flowing between widely distributed organizations.

The organization and smaller business processes of the public administration are very well structured and defined. Typical characteristics are exact job descriptions, task characterizations and allocations of responsibilities. These processes are similar to processes in other office automation environments (e.g. vacation requests or accounting of official journeys). Nevertheless, there is a vast number of processes that need adaptive support. Especially in complex and long-running processes, activities have to be managed in explicit planning phases which have an impact on subsequent process parts. Because of the strictly defined rules describing how activities have to be performed, processes get very complex and modeling of all exceptions gets too much expendable. Within these complex processes we can find a lot of unstructured parts which cannot be planned, defined, or automated ahead.

Some complex subtasks have to be fulfilled by cooperative groups or external organizations. Long running processes often contain decision points and need process refinement during execution. Additionally, some pretended structured and active processes need to be modified because of new restrictions, decrees or decisions of the legislature.

2. 2. General Requirements

Most systems focus on special application dependent requirements, but using WFMS as a strategical information system in a whole enterprise asks for further requirements. The difficulty is to provide process support for very heterogeneous processes, users, and technical environments. The following requirements are not intended to be complete but will be helpful to summarize some major challenges.

Due to the diversity of processes, there are different requirements for task coordination and cooperation within and between activities. Besides lightly structured workflows (like most administrative processes), semi-structured workflows (like production workflows) need to be supported as well as highly creative and unstructured process parts (e.g. ad hoc workflows). In addition, some cooperative process parts need synchronous telecooperation or coordinative group support.

Another challenge is to provide appropriate user interfaces for different users. The way users expect to work with WFMS depends a lot on their skills (especially in electronic data processing), their background knowledge and specific application areas. This problem will get the more important, the more users have to interact with the WFMS (e.g. actively participating in the process by performing workflow modifications).

Last but not least the technical environment of an enterprise is relevant for its system architecture and its functionality. Generally, a variety of computer systems, networks and software is used in large enterprises. Thus, the importance of requirements like distribution, interoperability, scalability, integration of applications, and legacy systems will increase rapidly. Furthermore, while not needed in each individual application scenario, specific functionality (e.g. security, consistency or reliability) must also be ensured in enterprise process management systems.

2. 3. Coordination and Control of Unstructured Workflows

As mentioned before, in some cases workflow relevant information related to unstructured processes is not available during buildtime. To perform processes the way desired, additional data from outside of the system has to be considered. While some missing information can be gathered automatically, interactions of actors also have to be considered. The following examples will summarize some reasons and requirements of workflow adaptations during runtime.

- **Determination of relevant data during execution**
  Execution of workflows often depends on dynamic information being determined during runtime. All systems use routing based on conditions or predicates. Occurrence of internal and external events which can be evaluated automatically, must also be considered. Some information which is well known to be required can be integrated by predefined user interactions, e.g. explicit decision points where an actor has to decide about actual routing or an explicit assignment of tasks to actors respectively roles.

- **Refinement and limited adaptation of predefined workflows**
  In some cases, limited unstructured process parts are not well known in advance (like optional activities or dependencies of other processes) or processes might contain explicit planning activities influencing the further execution (e.g. project plans).
  If occurrence of these situations can be anticipated, predefined activities should be inserted where refinement of subworkflows, insertion of additional activities or adaptation of some parameters can be performed. This feature can also be used for intentionally open process specifications where actors of predefined tasks should get the authority to perform them in their own way and under their own response. Sometimes in very unstructured processes, next steps cannot be planned anyway (e.g. scientific workflows), whereas evolutionary refinement of the process is highly recommendable or unavoidable.

- **Workflow modifications**
  A certain non-deterministic behavior can be achieved by these predefined adaptation mechanisms. But other situations will occur that cannot be foreseen. Due to unexpected external events affecting the process or due to faulty or deficient workflow modeling, the predefined workflow model needs to be adapted or modified. Therefore, a mechanism to perform unspecified workflow modifications on active workflow instances must be available.

3. Adaptive Process Control

Typically, we expect workflow systems to assign specified tasks to actors, controlling the underlying structured process. Information which is required to perform a workflow instance will typically be administrated exclusively by the workflow management system. The main idea of the following proposed solution is to develop a framework for flexible workflow management by offering sufficient adaptive support to handle unstructured processes, too.

3. 1. Flexible and Powerful Task Coordination

One solution to handle adaptations is to avoid them. Combining very flexible execution mechanisms with powerful modeling techniques will reduce the need for adaptations. As mentioned before, one major reason for adaptations is that modeling of very complex control flows regarding lots of exceptions will be highly expendable or impossible (for various systems). Furthermore, adaptations which can be particularly anticipated can be supported automatically.

Some simple concepts are inherent to most of the existing workflow management solutions, e.g. role resolution during runtime to determine an appropriate actor or control flow depending on predefined conditions on workflow relevant data. Some other mechanisms like event models and advanced control structures are hardly found in commercial solutions. Process support in SWATS is based on a flexible workflow engine with the following characteristics:

- Apart from prescriptive control structures, descriptive and complex structures are supported.
- Internal, external, and temporal events are integrated and included into conditions and predicates. Also know edge about history (e.g. assigning tasks to designated actors) or causal aspects should be included in order to control flexible workflows.
- Specific organizational procedures (e.g. delegation, check-back, hold-files etc.) and predefined adaptations (de cision points, ad-hoc refinements etc.) are offered to avoid difficult manual adaptations.
- Advanced organizational support (roles, representative rules, and dynamic roles etc.) can be used.

3. 2. Workflow Modification Services

A flexible workflow model should offer the opportunity to modify single workflow instances. Therefore the engine must offer interfaces to adapt workflow instances without affecting the workflow definition or other active instances. These adaptations can be related to different aspects of the workflow model. Most of the required changes refer to behavioral, informational and organizational aspects of workflows (Jablonski, 1994).

Modifications of behavioral aspects might be very limited operations like insertion of new activities, skipping/ deletion of predefined activities, or advanced start of activities with quite clear semantics. In addition complex modifications will be supported, e.g. insertion of subworkflows, restructuring of dependencies or modification of conditions and events in routing decisions.

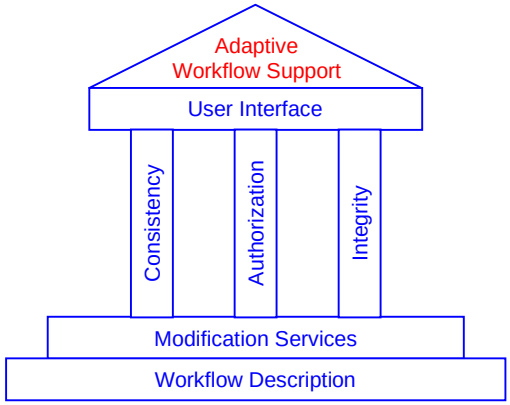

Fig.1 Solution Approach for Adaptive Workflow.

In contrast to behavioral aspects, the variety of modifications concerning organizational aspects is quite poor. Nevertheless, in lots of cases there is a need for manual changes of predefined roles or actors. Other powerful mechanisms will be offered by an organization service (e.g. representative rules or workflow relevant roles).

The informational aspect of a workflow defines what information is produced and consumed by processes. In this area modifications during workflow execution are required to add information or documents to activities or to perform changes of the data and document flow.

3. 3. Workflow Modification Control

Modifications of active workflows can be really problematical. Sometimes the semantic of modifications is difficult to evaluate and there is a danger of inconsistencies or side effects. Furthermore, it is very difficult to decide whether a modification is helpful and desired or not. The same adaptation, e.g. skipping the next activity (signature of the project manager), might be necessary in certain circumstances but fatal in others.

Therefore, manual modifications have to be controlled by several mechanisms (Figure 1).
By definition of advanced adaptation rights, modifications can be enabled or disabled. It has to be checked whether a special actor has the right to perform a particular adaptation in a given workflow instance, depending on its state.

Because typical workflow systems are not intended to be adaptive, consistency of adapted workflows has to be guaranteed. Apart from locking workflow instances that are currently under change, consistency rules must be defined and modifications have to map workflows from one consistent state to another.

Consistency checks will reflect some static rules, that each workflow has to guarantee. Another mechanism to control workflow modifications are integrity checks. User defined integrity rules can be modeled for each workflow definition and will be proved whenever a workflow instance is modified.

3. 4. Advanced Workflow User Interface

In some systems, workflow instances can be edited by administrators in special script languages but an adaptive workflow system should also enable workflow participants to make modifications and adaptations in a practical way. Because of this, users must be supported by comfortable graphical workflow

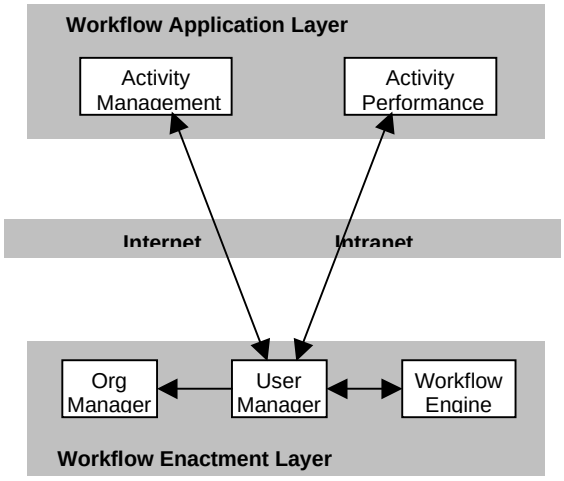

Fig. 2 Basic Components and Layers of SWATS.

editors which are able to present processes in a comprehensible way. Within this editor simple and complex adaptation services must be offered, corresponding to users' skills.

Additionally, complex adaptations can be supported by libraries of predefined adaptations and frequently used activities.

4. Adaptive Workflow Architecture

In the PoliFlow project an architecture has been developed to build a WFMS that meets the requirements of heterogeneous technical environments and heterogeneous user groups.

Due to requirements concerning unstructured process support, further components have been added to perform complex control structures and adaptation services.

4. 1. The SWATS-Architecture

The architecture of SWATS is characterized by two layers, the workflow enactment and the workflow application layer (Figure 2).

In the *Enactment Layer*, a commercial product could be used as a highly flexible workflow engine offering several interesting technical features (e.g. interoperability, scalability, performance). The engine is able to perform business processes according to a very flexible process model including complex rules and events. Being based on CORBA, it was well suited to the integration into heterogeneous environments (HP98).

A user manager and organizational services have been added to this engine. The User Manager administrates all scheduled activities with their possible actors and coordinates offering, assignment and termination of activities. Actor roles specified in the workflow description will be resolved by an Organization Manager. Both components are implemented in C^{++}.

In the *Application Layer*, user interfaces have been extended or built to enable adaptations from the client side. Activity management can be performed by worklist functionality and workflow enactment services can be used explicitly via CORBA or implicitly via HTML-forms including CGI-scripts. As

updates of the intray can be initiated according to a push and pull model dynamic changes of organizational aspects can be reflected very well. Apart from administrative operations (e.g. login, setup etc.) activity execution will be handled in the application layer.

Due to several requirements, all workflow applications are implemented entirely in Java and can be run in a WWW-browser. Moreover, using a framework for administrative, form-based activities workflows can be managed and activities can be executed without any further requirements of client software.

4. 2. Adaptive Support Layer

Extensions of the SWATS architecture have been made in order to enable unstructured process support. In an additional layer coordination of complex control structures on the one hand and control of workflow instance modifications on the other hand are being performed. The solution presented in this section should be applicable for most systems, hence it can be used as a reference model for adaptive WFMS. Enhanced components must be integrated in the architecture and will be discussed in the following.

Complex and descriptive control structures will be managed by the *Activity Coordination Service*. Due to functionality and the description model (e.g. petri nets) of the underlying engine, enhanced or customized control structures have to be defined. Once scheduled they will be controlled by this service interacting with the User Manager and the workflow engine. Thus, structures like delay, deadline or sequential row will be coordinated according to the interactions of involved users.

If manual modifications are required, workflow instance information will be extracted from the engine to build up an internal representation, where adaptations can be performed and controlled properly. *The Workflow Instance Manager (WIM)* offers a defined set of workflow adaptations which are first managed in the internal representation before being propagated to the entire workflow enactment system.

All manual adaptations will be controlled by the *Modification Control Manager*, which has to ensure syntactical and semantic correctness of workflows.
In most systems, syntactical consistency of workflows will be proved during buildtime (e.g. by workflow modeling tools) and ensured by defined state transitions of the engine. Because buildtime and runtime cannot be strictly separated in cases of manual modifications, consistency of workflow instances has to be controlled by explicit checks.

According to the workflow definition, it will be examined whether users have the authorization to perform applied modifications or not. Therefore, expendable analysis of the workflow instance may be required to evaluate complex dependencies and conditions concerning the current state of a workflow instance.

Finally, the Modification Control Manager has to guarantee integrity of adaptive workflows. Therefore, user defined integrity rules of affected subworkflows are re-evaluated to ensure semantic correctness of workflows, even in case of manual modifications.

Apart from control mechanisms, functionality must be provided to ensure atomicity, consistency, and durability of workflow adaptations in distributed environments. Because conflicts of workflow modifications are not expected to occur very frequently and duration of performing modifications is not critical, usual locking mechanisms can be applied.

4. 3. Impact of Adaptive Support on WFMS

This architecture has been developed to be applicable in different WFMS. Nevertheless, there are impacts on the WFMS which have to meet several specific requirements (Figure 3).

The engine must offer sufficient interfaces for instance handling (e.g. export interfaces, instance modification interfaces, and interfaces for locking etc.) and enough flexibility to integrate complex activity

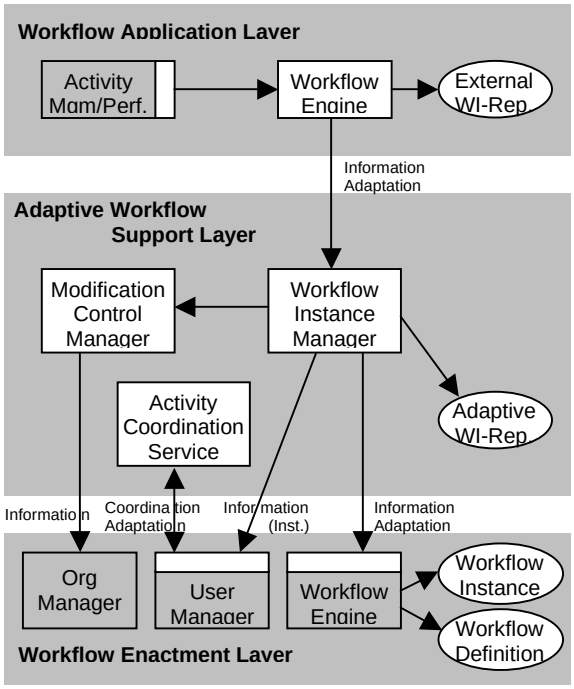

Fig. 3 An Adaptive Support Layer for WFMS.

coordination (e.g. state model and interfaces). Also, the workflow model has to be extended by definition of adaptability aspects (adaptation rights) and workflow integrity rules.

Depending on the functionality of used components, enhanced properties and interfaces may be required to perform necessary adaptations not only in the engine, but also in other enactment components (e.g. worklist handler). Concepts of the SWATS User Manager are adequate to these requirements because worklist management can be handled in a very flexible way.

Finally, any workflow participant must be able to use adaptive support functionality. Approaches where very special skills are necessary to perform adaptations are insufficient because of a variety of user profiles in large enterprises. Thus, the workflow application layer in SWATS is complemented by a graphical workflow editor and a library service. As mentioned before, comprehensibility of process models may depend on various factors, whereas different external views on workflow instances have to be offered. As a first step, very structured process views are offered similar to structured programming concepts. Within this views of the workflow editor, actors are able to adapt workflows manually .

5. The User Interface for Workflow Adaptations in SWATS

The graphical workflow editor in SWATS offers multiple functionality to different user groups. On the one hand it can be used by specialists as an usual workflow designer to view, build, or modify workflow definitions. The engine supports version control mechanisms of workflow definitions and offers read/write interfaces. These original workflow definitions will be parsed by the WIM and be transformed to an internal process model.

On the other hand all process participants can use the editor for viewing and editing single workflow instances. When loading instances, the workflow definition will be complemented by the corresponding instance information, given by the engine and the User Manager.

5. 1. Requirements and Concepts

There are three requirements in particular which have had great impact on the design of the workflow editor and the underlying WIM: comprehensibility and usability of models and tools, transferability of concepts and tools, and applicability of models and concepts to support adaptation control mechanisms.

To enforce architectural and conceptual independence all components of the adaptive support layer are working on a specific internal process model. The control flow of this model depends on an extensible set of control structures. This approach has been chosen because most of the existing process models can be easily extended and transformed into a structured form. Furthermore effects of adaptations on structured processes can be analyzed very well, which simplifies the control of workflow instance adaptations. Structured modeling also encourages modularity, abstraction, and comprehensibility of processes.

The WIM offers a defined set of basic and an extensible set of complex operations to modify workflow instances. To increase usability of the editor these operations can be performed in an easy way by operating graphically (like drag and drop), by editing object properties, or by selecting predefined complex adaptation procedures.

5. 1. Workflow Aspects and Editor Views

Even if the adaptations themselves can be performed easily, actors must get a general idea of what the process is doing and how they can adjust it in an appropriate way. In order to gain these insights, workflows are presented in several views which belong to different aspects of workflow models:

- A behavioral view focuses on the control flow, showing control structures and their accessory activities or subworkflows.
- A informational view gives an overview of the use of workflow data and outlines the flow of application data between activities. For complexity reasons, it is configurable which data shall be viewed or not.
- A temporal view organizes the activities by time zones and gives a good impression of when activities have been executed or when they will be started/finished in the best/worst/average case.

Each view has a special intention. Nevertheless, some aspects can not be strictly separated (e.g. functional and behavioral aspects) or do not need a separate view (e.g. assignment of roles to activities).

As a result, the presentation of activities (illustrated in rectangles) has been made configurable by including the most important additional information like work node and activity name, role/actor, state (by color or text), start/end date, or duration.

Figure 4 shows the editor with a loaded workflow instance of a typical booking process shown in the behavioral view. Adaptations can be performed by the pull-down menu *Modify* or by using the vertical toolbox. This box contains icons for all basic structural operations (e.g. insert, delete, cut, and paste).

Figure 5 shows the same instance in the informational view. The flow of application data is visualized by colored, labeled arcs, which can be picked out to get further information. In this view, control flow information has been reduced to explicit predecessor-successor relations represented by dashed black arcs.

Since some information can be better represented textually another view has been added, the object view. When selecting workflow objects (e.g. activities, control nodes, arcs), an additional window with object properties appears. The object view is divided into several parts related to the different workflow aspects. Depending on the editor context (e.g. the current active view), the object view will pop up with the adequate part.

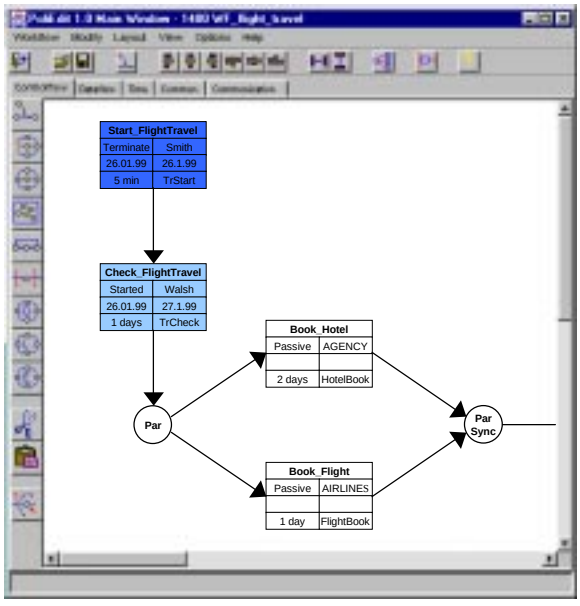

Fig. 4 Adaptation of functional and behavioral aspects.

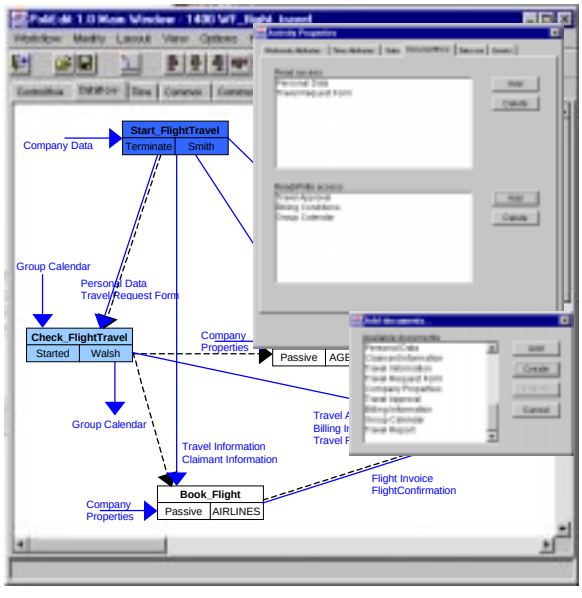

Fig. 5 Adaptation of informational aspects.

Data flow in SWATS is implicitly defined by rights of activities to data. That is why modifications of data flow must result in modifications of the use of data. Apart from graphical adaptations, this can be done by activating the object view and modifying the table with data access rights.

Other parts of the object view belong to temporal properties of activities, organizational dependencies or path conditions in control structures.

5. 2. Architectural and Implementation Issues

The editor has been implemented in Java according to the architectural principles in SWATS.

To ensure independence from the WIM, the editor works on a local model which need not necessarily be the same as the WIM-model. Furthermore, the editor has been built according to the model-view-controller principles. Since every application domain or user group has its own user specific requirements, a single graphical representation of workflows cannot be sufficient for enterprise-wide workflow management. Due to this fact, the editor has been designed in a way that other representations can be developed by re-using design and code.

The layout of graphical process presentations is an important issue to enforce comprehensibility. Due to the complexity of this problem, which is beyond the scope of this work, just a simple automatic layout algorithm has been implemented. But the layout of each workflow instance view can be arranged independently, supported by the given layout operations from the horizontal toolbar. As a consequence, the workflow instance model had to be extended by layout information.

Due to the expendable graphical computations and the well known performance of JAVA, the system requires an adequate memory and computing power (e.g. Pentium 300 with 64 MB RAM for acceptable performance).

6. Current State and Future Work

Both basic layers, the workflow enactment layer and the workflow application layer, have been developed in the PoliFlow project. Depending on a special framework which supports form-based administrative processes over Intranet or the Internet, some workflow applications have been realized for the customer.

Due to conceptual and technical problems a prototypical approach has been chosen to develop the integrated adaptive support layer.

In a first phase, basic concepts of the layer's functionality have been acquired and proved in rapid prototypes; first versions of the Instance Manager, the graphical Workflow Editor, and the Modification Control Manager have been implemented.

Because of conceptual enhancements and technical issues (e.g. new versions of the workflow engine and the software environment), these prototypes had to be re-engineered. Implementation of the second version of these components has been finished as well as the implementation of the Activity Coordination Service.

Actually, conceptual work is related to the area of consistency and integrity control, to expand the Modification Control Manager. Further implementational work will be spent on the integration of the commercial workflow engine into SWATS (e.g. advanced modification interfaces and workflow instance locking).

As a first complete prototype the system has been presented at CeBIT'98.

7. Conclusion

The intention of this research is to point out certain limitations of existing WFMS combined with a feasibility study of new and demanding workflow concepts. The results have to be examined and evaluated in the application field where continuous interactions with selected customers are guaranteed. Due to the very heterogeneous application environment within the project, the results are expected to be suitable for many other enterprises and WFMS.

Next generations WFMS should not subsume the wide area of groupware applications like synchronous telecooperation services or groupware platforms supporting complete unstructured work. Nevertheless,

there should be a certain quality of integration, offering more than simple encapsulation of unstructured tasks in special activities.

The proposed solution and reference architecture presented in this paper should be one way to achieve aseamless integration of structured and unstructured processes provided by powerful workflow technology.

6. References

Amberg Michael, 1996, "Modelling Adaptive Workflows in Distributed Environments". In Workshop on Adaptive Workflow, Proc. of the First Int. Conf. on Practical Aspects of Knowledge Management (PAKM), Basel.

Cichocki A., Helal, A., Rusinkiewicz M. Woelk D., 1998, "Workflow and Process Automation - Concepts and Technology". Kluwer Academic Publishers, Boston.

Dogac A., Kalinichenko L., Oezsu M.T., Sheth A., 1997, "Advances in Workflow Management Systems and Interoperability". NATO Advanced Study Institute.

Hewlett Packard, 1998. "HP Changengine: Dynamic Solutions for IT/Business Alignment". White Paper at http://www.ebizsoftware.hp.com/itmngmt/changeng.html

Jablonski Stefan, 1994, "MOBILE - A Modular Workflow Model and Architecture". In Proc. of the 4th Int. Conf. on Dynamic Modelling and Information Systems, 1994, Netherlands.

Jablonski S., Stein K., Teschke M., 1997, "Experiences in Workflow Management for Scientific Computing". In 8th International Workshop on Database and Expert Systems Applications (DEXA), France.

Kindler Thomas A., 1997, "Activity Based Security in Intranet and Internet Workflows". In Proc. of the 7th Annual Workshop on Information Technologies & Systems.

McCarthy D., Sarin S., 1993, "Workflow and Transactions in InConcert". IEEE Bull. of the Tech. Com. on Data Engineering, Vol. 16, No. 2.

Reichert M., Dadam P., 1997, "A Framework for Dynamic Changes in Workflow Management Systems". In 8th International Workshop on Database and Expert Systems Applications (DEXA), France..

Sheth Amit, 1997, "From Contemporary Workflow Process Automation to Adaptive and Dynamic Work Activity Coordination and Collaboration". In 8th International Workshop on Database and Expert Systems Applications (DEXA), France.

Siebert Reiner, 1996, "Adaptive Workflow for the German Public Administration". In Proc. of the First Int. Conf. on Practical Aspects of Knowledge Management (PAKM), Workshop on Adaptive Workflow, Basel.

Siebert R., Kindler T., Soyez, T., 1997, "Integrated Workflow and Telecooperation Support for the German Government". In Proc. of the Symposium on Applied Computing (SAC), San Jose.

Wainer J., Weske M., Vossen G., Medeiros C., 1996, "Scientific Workflow Systems". In NSF Workshop on Workflow and Process Automation in Information Systems, Athens-Georgia.

Weske Mathias, 1998, "Flexible Modeling and Execution of Workflow Activities". In Proceedings of 31st Hawai'i International Conference on System Sciences, Software Technology Track (Vol VII), IEEE Computer Society Press.

Wolf M., Reimer U., 1996. Proc. of the First Int. Conf. on Practical Aspects of Knowledge Management (PAKM), Basel.