ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Rainer Manthey and Viacheslav Wolfengagen (Eds)

Advances in Databases and Information Systems 1997

Proceedings of the First East-European Symposium on Advances in Databases and Information Systems, (ADBIS'97), St Petersburg, 2-5 September 1997

Motown: a Practical Approach to Workflows

Jean-Loup Chappelet, André Le Grand, Michel Prevel, and Jean-Jacques Snella

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

ISBN: 3-540-76227-2

Motown: a Practical Approach to Workflows

Jean-Loup CHAPPELET, André LE GRAND*, Michel PREVEL

Institut de Hautes Etudes en Administration Publique

21 route de la Maladière

CH-1022 Chavannes-près-Renens (Lausanne)

Jean-Jacques SNELLA

C-Log SA

3, rue du Léopard

CH-1227 Geneva

Abstract

Workflow management systems (*WFMS*) are intended to improve business processes in terms of analysis and of automation. Our approach, based on a realistic development life cycle, indicates the modeling and implementation requirements to speed up the development of workflow applications (mainly administrative and ad-hoc workflows [5]). This paper presents both our methodology and tool to support this lifecycle for human-oriented workflows [8].

1 Introduction

Diagram 1 shows a realistic development life cycle for workflow applications which takes into account the workflow evolution that will undoubtedly occur.

Figure 1: workflow development life cycle

Each life cycle steps are described below:

* Also at Information Systems Department, University of Geneva

1.1 Business process elicitation

This step is to identify and understand the processes of an organisation. This operation leads to mental models shared by all the people involved at this stage. The mental models are the organisational analysts' mental representation of the processes. Knowledge acquisition methods like those developed by expert system designers are highly suitable here.

1.2 Business process modeling

This step is to formalize the mental models in a conceptual model used by designers to perform analysis. We use three levels of process modeling (Cf table 1): *abstract, descriptive and operational*.

The mental models are represented at the abstract level to model the objectives of business processes. The descriptive level describes the means involved to fulfill these objectives. To build a workflow application, the descriptive models are refined into operational models. The operational models of processes reflect exactly how the workflow application will work.

1.3 Workflow analysis

This step is to test the workflow specification and often highlights modeling errors and misunderstandings about business processes or between the analysts. The analysis may change the perception and objectives of the processes. This reconceptualisation may also modify either the abstract, the descriptive, or the operational models.

When the process representation does not evolve any more, a new series of tests is suitable, like bottlenecks detection or time/cost simulation, for instance.

1.4 Workflow implementation

This step is to transform operational models of processes into an executable system. To ensure an error-free transformation, it should be fully automated. This approach based upon rapid application development, enables quickly to work and test the application.

1.5 Business process re-design

According to the evolution of business processes, possibly coming from daily work with the application, re-design leads to some changes in the experts' mental models.

To support most of these life cycle steps, we have developed two tightly coupled tools: a graphical specification editor and a workflow enactment system. The specification editor is used to model processes and analyse workflows.

The implementation is fully automated, the target system is Lotus Notes¹. The workflow enactment system monitors the process execution.

The following sections describe these tools and how they help to manage the business process evolutions.

Since the workflow applications are built on top of a groupware tool, some cooperative work and workflows can be mixed.

¹ Lotus Notes is a registered trademark of Lotus Development Corporation

Level	Main concepts	Purpose
Abstract	(Sub)Function, Activity, Packet	modeling the objectives. What is to be done?
Descriptive	Procedure, Role, Resource, Facility, Operation, Actor, Team, Job Description	modeling the means. How? By who?
<i>Operational</i>	<i>Document, State, Constraint, Document Structure, Notification</i>	<i>workflow specification. What is computerized?</i>

Table 1: The Motown process modeling levels

2 The graphical specification editor

2.1 Business process modeling

Like many workflow management systems, our approach of modeling is activity-based [6]. The conceptual model is an extension of the OSSAD model. OSSAD (Office Support Systems Analysis and Design) [3] stems from an ESPRIT project and is widely accepted by those carrying out for large scale office work modeling (table 1). In our sense, OSSAD is widely accepted because it is not only a set of models, a modeling technique, but a method: OSSAD also provides a set of guidelines for modeling and managing a project [7, 2].

In order to specify workflows², Motown adds an operational model to OSSAD.

The *abstract level* (see for example fig. 1³, 2, 3) serves to specify what is to be done within an organisation. It is based on a breakdown of the organisation into sub-systems according to defined objectives (*functions* and *sub-functions*).

The sub-systems are related to each other and to the environment outside the organisation by a flow of data packets. At the finest level of detail, the sub-functions regroup a number of *activities*. As the name suggests, the abstract level disregards the support system used to reach the organisation's objectives.

The *descriptive level* concerns the organisational, human and technological support systems which are used to accomplish the organisation's objectives. This level contains two models, namely the *procedure model* and the *operation model*. The former describes how the information flows between procedures. The latter details a procedure (Cf fig. 4). It uses the concepts of role, operation, resource and facility (cf. glossary).

The *operational level* extends the procedure model (of a descriptive model) to specify what is computer-supported as a part of a workflow. For this purpose, we add the concepts of document state and document structure (Cf fig. 5). The constraint concept is also very useful because an actor usually carries out several roles. We define constraints between operations which apply during the same workflow instance. They are of two types, namely "must perform" and "cannot perform", for instance:

- the actor who performs the operation *Fill in* must also perform the operation *Modify* on the same document.
- the actor who performs the operation *Fill in* cannot also perform the operation *Approve*.

A set of operational models defines a workflow application which is implemented by Motown Builder.

²in table 1, the extensions of OSSAD for workflow are written in italics. At the end of this paper a glossary is provided.

³fig 1 is also an abstract model of a workflow development lifecycle.

Figure 2: abstract level: functions of an organisation

Figure 3: abstract level: sub-functions of the *Sales Function*.

 Figure 4: descriptive level: an operation model of the activity *Expense submission*

 Figure 5: operational level: a workflow model of the procedure *Expense submission*

2.2 Workflow analysis

The analysis, test and debugging of workflow specifications are mainly based on net analysis and simulation. Cross-referencing analysis is also used to maintain consistency between the three levels of modeling.

2.2.1 Net analysis

Full advantage has been taken here from the fact that the operational models can be translated into Petri Nets. Feasibility checks can then be performed using these net representations, in order to identify unusual and unexpected situations in the workflow, as early as at the specification. For instance, Motown checks the reachability of operations and states and the liveness of the graph which is the workflow model.

2.2.2 Simulation

Moreover, dynamic quantified simulations may be launched in order to answer numerous questions about cost, duration, and processes requirements for human or material resources. During the simulation, an animation displays the queues on procedure objects.

A simulation may concern the durations, the costs or both. The parameters of a simulation are:

- the assignments of actors to roles⁴,
- the probability for each OR post-condition ,
- the duration and cost of an operation⁵,
- the duration and cost of a resource. Here, the duration means the time to transmit the resource.
- the duration and cost of a procedure⁶,
- for an operation, a number of waiting inputs to start the operation, an earliest time and a latest time. These parameters are optional.

Any duration may be fixed or variable according to a *Gamma* law of probability whose average is user-specified. A cost can be broken in two parts: a fixed part and a variable part. The fixed cost is taken into account whenever an object⁷ is used. The variable cost is time-dependent.

The launching of a simulation requires to:

- specify the duration of a simulation (per unit of time),
- the average interval between two runs of a procedure. Like costs, this interval is either fixed or variable, according to a Gamma law.

For each procedure, the simulation report contains:

- the values of all parameters,
- the list of termination points of the procedure,
- the earliest, latest and average durations,
- the number of procedure's runs,
- the number of procedure's ended runs,

⁴Per unit of time, an actor performs only one operation.

⁵it includes the duration to create a resource or to update a document.

⁶these parameters are used when a workflow model includes another procedure

⁷an operation, a resource or a procedure

Level	Main concepts	Purpose
monitor	signal, query, document state	determine and control the change of document state
workflow meta-model	procedure, role, actor, operation, document type, state, notification, constraint	specify when and how operations affect documents.
information structure	document structure, forms and folders	specify the structure of document types
documents	documents	user-defined documents

Table 2: The levels of the Workflow Enactment System

- the minimum, maximum and average costs,
- the amount of treated operations, resources and documents by state,
- the amount of uncompleted operations, resources and documents by state,
- for each role, the number of actors effectively involved, and the average time of work of actors per role.

It is also possible to focus simulations on a particular termination point of a procedure. If a termination point represents a failure of a procedure, we may evaluate for instance, the costs of a failure.

An accurate quantification of parameters is required to run a realistic simulation. Defining a pertinent quantification is time consuming and involves a rigorous method to evaluate the durations and costs of operations. If such an effort is not carried out, the simulation results have no meaning. Another approach is to implement the workflow application and to use the traces' reports to specify the simulation parameters. A simulation, even if costly to run, is very useful to test the feasibility of procedures designs according to a set of given resources.

3 Workflow implementation

With Motown, the implementation step of the life cycle is fully automated (fig. 6⁸). When the workflow specification fulfills the designers' criteria, Motown builds a Lotus Notes base which contains all the processes definitions. The designer then adds manually the documents structures as Notes Forms. By assigning actors to processes' roles, the Notes base becomes a workflow application which is ready to deploy to all its users. workflow enactment system

4 The workflow enactment system

4.1 Architecture

The workflow enactment system is a Lotus Notes base including the Motown workflow engine. The four levels of its architecture are: monitor, workflow meta-model, information structure, documents (table 2). The monitor controls the evolution of a document according to the workflow. The workflow specification is stored according to a meta-model. The information structure defines the forms which allow access to the documents. The fourth level is the level of documents created, updated and deleted by the users. Lotus Notes covers the two bottom levels. The monitor updates the document status according to the user work. The updates depend on the specification workflow stored in the meta-model.

⁸ this figure is also a descriptive model of the implementation step of figure 1.

4.2 Activating and monitoring the flow

From the end users viewpoint, three main features of an application are central to the interactions with the workflow enactment system:

- *Graphical help*

All the graphical specifications of processes are available to users, either as an organisational documentation of the application or as an on-line contextual help, when they process a document.

- *To Do Lists*

They are used to display the documents which a user has to process. The displayed list takes into account the user's roles, the current state of the documents, and constraints between operations, if any. A workflow application may also contain other document views, as in any Notes Base.

- *The Go on action*

This action⁹ is present whenever a user edits a document for which he has to perform some operation.

Since we consider human-oriented workflows, not any document modification by an actor corresponds to the completion of an operation: the actor has to assert that he achieved some work on a particular document (some operation whose input state is the current state of the document). The action *Go on* of the document is to assert this completion. This action sends the assertion the monitor. The monitor asks information to the workflow meta-model in order to determine the achieved operation and its output state. This query may lead to interactions with the actor: several operations may share the same input-state, or the actor has to choose one of several possible output-states.

5 Managing the evolution of business processes

Several factors usually lead to an evolution of business processes. These factors are either external or internal to the process itself. Among the external ones, we may cite changes with the legal requirements and regarding organisation, as well as the emergence of new technologies standards (like EDI or internet) that are requested by customers or suppliers. Other reasons cited in [5] include increasing customer satisfaction and quality of products, as well as meeting new business challenges and opportunities. External factors are usually overriding and they lead to radical re-engineering.

Internal factors come from the daily use of workflows applications. For instance, users may have new ideas to increase the overall process efficiency. The major internal factor is the periodic reports of workflows execution which include workloads, operation completion times and so on. These statistics provide meaningful data to run accurate simulations on process changes. For instance: what is the average procedure completion times when we add actors to a role?

The users' feedbacks of a workflow system also highlight problems like unforeseen bottlenecks that result from either the design of the process itself, or unrealistic parameter quantifications. Internal factors lead to design improvements of the current processes. Iterative and fast (re-)design and (re-)implementation tools are thus very useful to such an evolutionary design approach.

Whatever the root cause of the process evolution, workflow enactment systems should provide tools to update several running workflows instances.

In a document / human role -oriented system like Motown, there are three dimensions of changes at the enactment level, which are usually combined in an evolution:

- *Changes on the relevant documents set of a process, and on document structures.*

⁹which is attached to a button

At the design level, if document types are removed from a process, then, at the enactment level, all the pending operations on documents of these types are stopped. The documents come out of the workflow and can be archived by the application manager.

For the changes on document structures, we rely on Lotus Notes which provides the simple form/document model where the form does not define the structure of a document.

- *Changes on actors, roles and assignments.*

At the enactment level, Motown provides a diagnosis tool for the detection of access rights violations on documents. Access rights violations may occur only if workflow models include constraints between operations.

- *Changes on the flow.*

They result from a new design of procedures. Some procedures/operations may have been created, other removed. Their ordering may be different. The assignment of role to operations should also be modified. Motown uses a diagnosis tool to detect some blocking situations which may occur if a workflow model includes constraints between operations.

In order to satisfy the constraints on "ill" documents¹⁰, the organisational analysts have to understand the effects of changes on the current workflow instances.

Although each modeling operator is conceptually well understood, their overall effects are more difficult to understand [3]. One difficult point is to decide, which documents in which state have to follow the "old" workflows, whereas others have to follow the new process definitions (immediate change). Moreover, some documents will have to follow the old workflow until they reach a particular state, and then they will follow the new processes (delayed change). How to recognize automatically these situations is still an open issue, thus, our workflow enactment system provides functions to make these changes when they are required.

6 Conclusion

According to [1] which provides meaningful arguments, we claim that state-based models are very pertinent for workflow applications. In Motown, the underlying Petri net representation is very useful for analysis and simulation of the process models. Our approach is practical and improves current workflow methods or tools on the following topics:

- it is based on a more realistic lifecycle which integrates the business process evolution as a possible step,
- the modeling concepts are easily understandable by project teams and end users, cutting down the training and acceptance time.
- simulations provide realistic and practical results.
- implementation is fully automated.

Modeling changes can be integrated in an existing application either automatically (for simple changes) or assisted by an analyst (for more complex changes).

7 Perspectives

This last section presents some issues that should be addressed when tending to improve business process modeling and re-design.

¹⁰A document is said to be ill, if its history prevents the new procedure specification (whatever the changes are) from being satisfied according to the current role assignments.

7.1 Improvement of the analysis

The analysis of business processes focuses on testing only whether the specification matches the users' needs. For some critical applications, we may want to prevent the occurrence of some undesirable actor coordination or situation, such as the synthesis of distributed critical information, or an unrecoverable situation resulting from a particular flow. Artificial Intelligence Planning tools may be very useful for improving analysis of undesirable situations.

7.2 Process evolution analysis

To analyse the process evolution, a method should enable the evaluation of the changes' benefits and should guide the transition to the new processes. The evolution itself should be guided according to the results of simulations or enactments.

7.3 Procedure canceling

Another important issue is to manage the cancellation of a procedure. This issue is meaningful at the modeling and implementation steps. The modeling level should provide concepts to specify which, when and how a procedure can be aborted. Moreover, the consequences of a procedure cancellation on other procedures should also be analyzed. At the implementation level, the issue is to define a powerful mechanism ensuring correct management of the task cancellation. This issue might be a research issue.

7.4 Mixing unstructured and structured processes

At the enactment level of Motown, Lotus Notes enables both cooperative work and structured work.

At the modeling level, this cooperative work could be specified according to some well known cooperation models. The interactions between these works types should also be modeled.

References

- [1] Van der Aalst W.M.P. Petri-net-based workflow management software. In: <http://lsdis.cs.uga.edu/activities/NSF-workflow/finaldocs.html>, 1996
- [2] Cha ppelet J-L, Snella J-J. Un langage pour l'organisation: l'approche OSSAD. Presses Polytechniques et Universitaires Romandes, Lausanne 1997
- [3] Conrath DW, Dumas P et al. Office Support Systems Analysis and Design: a Manual. Esprit Project 285. IOT, Munchen 1989
- [4] Ellis C, Rozenberg G. Dynamic change within workflows systems. proceedings of the international Conference on Organizational Computing Systems", 10-21, october 1995
- [5] Georgakopolous D, Hornik M, Sheth A. An overview of workflow management: from process modeling to workflow automation infrastructure. Distributed and Parallel Databases, 3, 119-153, 1995.
- [6] Marshak R. Software to support BPR- the value of capturing process definitions. Workgroup Computing Report, Patricia Seybold Group, vol. 17, no. 7, july 1994
- [7] Sherwood-Smith MH. An Introduction to OSSAD. <http://www.unil.ch/idheap/mss.htm>, 1993.
- [8] Sheth A & al. Report of the NSF workshop on workflow and process automation in information systems. <http://lsdis.cs.uga.edu/activities/>, 1996.

A Glossary of motown modeling concepts

This glossary includes the definitions of concepts from OSSAD [7] and Motown. Concepts of the operational level A.3 are ours.

A.1 Abstract level

Function (or subfunction) A major division of the actions and of the structure of an organisation, usually based on the need to co-ordinate, direct and control to obtain specified outcomes independently of the actual means used. A *function* may be divided into *subfunctions*, with the number of such levels reflecting the need for detail.

Activity The finest division of *functions*. It is the smallest amount of work that must be done for which a meaningful objective and/or easily definable output can be identified.

Packet A set of objects/data that passes between activities and between them and the external environment.

A.2 Descriptive level

An *activity* in the *abstract level* maps onto a *procedure* in the *descriptive level*. For instance the activity *workflow implementation* (fig. 1) maps on to the procedure *workflow implementation* detailed in figure 6.

A packet in the *abstract level* maps onto a set of *resources* in the *descriptive level*. For instance, the packet *workflow specification* (fig. 1) maps to the resources: *process definitions*, *documents structures* and *actors* (fig. 6).

Operation The basic (most detailed) element work relevant to the description of the organisation. The desired level of detail depends on the nature and purpose of the specific model.

Procedure A coherent set of *operations*, undertaken by one or collaboratively by several *roles*, which result in a meaningful objective being achieved or a the production of *resources*.

Resource Data or objects which are inputs to or outputs from *operations/procedures*.

Facility Physical or technological support to perform work.

Role An organisational responsibility covering a set of *operations* performed by an individual or more than one individual if they are collaborating in a *procedure*.

Team An aggregation of *roles* based on one or more organisational requirements for co-ordination or control. *Teams* may be aggregated into higher level *teams*, culminating at the level of the entire organisation.

Task A *task* is a term that is occasionally used in the OSSAD context. It is not represented as a symbol in the OSSAD formalism. It is the set of *operations* which are performed by a given *role* and contained within a single *procedure*.

Actor An individual who carries out a *role*.

Job description A set of *roles* an *actor* has to carry out.

A.3 Operational level

The *operational level*, a refinement of the OSSAD descriptive level is to specify workflows. The concepts introduced at this level are specific to Motown. For each *procedure*, there is a *workflow model*. Defining a workflow model of a procedure consists of refining the operation model of the procedure. Thus the question is: which *resources* will be *documents* in the workflow enactment system? Then states are added to documents to specify how *operations* affect the *documents*.

Document a workflow-supported *resource*.

State the *document* status within a *procedure* execution. It specifies how an *operation* affects the *document*.

Document structure A set of fields whose domains are the same as those of Lotus Notes.

Notification A signal sent to each *actor* of a *role*. It signals that a document reaches a state that enables an operation of their task to be carried out.

Constraint a rule between two *operations* stating whether the same *actor* should or should not perform these two *operations* in the same *procedure* execution.

Figure 6: workflow implementation procedure