Dynamic Changes in Workflow Participant Assignment[1]

Mariusz Momotko[*] and Kazimierz Subieta[+#]

[*)] Rodan Systems S.A., Warsaw, Poland
[+)] Institute of Computer Science PAS, Warsaw, Poland
[#)] Polish-Japanese Institute of Information Technology, Warsaw, Poland

Abstract. Workflow management systems (WFMSs) need to adapt dynamic process modifications. In current WFMSs the scope of dynamic modifications is mainly focused on control flow, while other dynamic aspects are neglected. In this paper an approach to adapt dynamic modification in workflow participant assignment (WPA) is presented. The approach extends the meaning of WPA that is proposed by the Workflow Management Coalition. The extension covers dynamic aspects and expresses complex relationships between control, audit and relevant data. On basis of the new definition a WPA Language (WPAL) is proposed. WPAL is a programming interface, which makes it possible to assign dynamically workflow participants. WPAL has been implemented in *OfficeObjects® WorkFlow* and deployed among several major customers of Rodan Systems. The paper also presents implementation results.

1 Introduction

A desirable feature of the workflow technology is adaptability to workflow changes that can occur during a workflow execution (referred further to as *dynamic modification*). Dynamic modification is important for real-life workflows that need to be changed frequently to adapt to changes of the environment. Such real-life instances might include department reorganisation, absence of a workflow participant, unavailability of a printer, etc. Dynamic modifications, depending on their durability, is one of the following types:

- **Modification of an executed workflow** (referred further to as *workflow instance*). The modification concerns some workflow instances and usually is connected with transient changes or exceptions.
- **Modification of a workflow definition**. It concerns all workflow instances and usually is connected with workflow optimisation.

[1] This work was supported by the European Commission project ICONS, project no. IST-2001-32429.
Paper accepted to *Sixth East-European Conference on Advances in Databases and Information Systems, ADBIS'2002*, September 8-11, 2002, Bratislava, Slovakia.

Information on how a workflow has to be modified can be achieved on the basis of:

- *control data* representing the dynamic state of workflow instances and the WFMS (e.g. workflow definitions),
- *audit data* representing the history of workflow instances execution,
- *relevant data* used for evaluation of conditional expressions, for instance, expressing transitions or participant assignments.

Currently some WFMSs support dynamic workflow modification, including both workflow instance and workflow definition. The research and development on workflow dynamic modifications is focused on the control flow modification [1, 2, 3, 4, 5, 6, 9, 10, 11, 12]. Dynamic modification of other aspects of workflow, especially workflow participant assignment (WPA) is usually omitted. The ability to adopt dynamic changes of participant assignment seems to be vital for organisations where processes are customer-oriented.

In this paper an approach to dynamic modification of WPA is presented. The Workflow Management Coalition (WfMC) definition of WPA has been extended to cover dynamic aspects expressing complex relationships between control, audit and relevant data. On the basis of the new definition a WPA Language (WPAL) is proposed. WPAL makes it possible to express complex dynamic aspects of workflow assignments. It also introduces a reuse mechanism within WPA definitions. WPAL has been implemented in *OfficeObjects® WorkFlow* (*OO WorkFlow*) [7, 8] and deployed among several Rodan Systems' major customers.

This rest of the paper is organised as follows. In section 2 a framework for the new approach is presented. This section introduces a new meaning of workflow participant assignment and defines WPAL. Section 3 describes an implementation of WPAL in *OO WorkFlow* and presents practical results. Section 4 presents a summary and further plans.

2 A Framework for WPAL

Rodan Systems, a leading Polish developer of document management systems, collected requirements for modern WFMSs that could support real business processes. The requirements have been influenced by needs of existing and potential customers having experience on use of WFMSs and on formal description of processes.

On the basis of the requirements a list of main features that a modern WFMS should support has been prepared. One of the most viable features is the ability to modify workflows to adapt dynamic changes of workflow environments and changes of WPA. Especially for organisations that are customer-oriented dynamic changes of teams to serve a given case is a key issue.

2.1 Requirements for a Modern WFMS

A WPA should be able to express the following aspects:

- Complex relationships on:

  - Control and audit data – data on finished or currently executed workflows. For example: a person that has the lightest workload or minimal number of tasks to perform, a workflow participant that started the workflow, a workflow participant that performed the previous/preceding activity, a salesman that in the last week performed more than 30 workflows.
  - Relevant data – processed data, organisational structure or other data. For example: a participant that is defined as a tester of a given system bug, an employee that is the supervisor of John Smith, a person that knows Java and XML, a workflow participant that has the 'knows English' role.
- The situation when workflow participants assigned to a given activity are selected ad-hoc, manually during workflow execution.
- Organisational and functional structures, in particular user groups that exists in an organisation.
- The situation when exactly one workflow participant from a selected group should perform an activity,
- The definition of a workflow participant who will perform an activity if workflow participant assignments return inadequate set (e.g. an empty set).

To assure flexibility of our WFMS we decided that its architecture, interfaces and workflow definition language are to be compliant with the WfMC standard. However, it seems that the issues concerning WPA are not well defined in the standard and too limited in existing WFMSs. Thus we put stress on developing these mechanisms.

2.2 Problems with WfMC Definition of WPA

According to the WfMC's definition [13, 14, 15], a WPA should define explicitly the set of participants that will perform workflow activities. A participant is one of the following types: resource set, resource (specific resource agent), organisational unit, role (a function of a human within an organisation), human (a WFMS user) or system (an automatic agent). A human and a system are static elements of the workflow participant definition and are known at the workflow design time. A resource, an organisational unit and a role are dynamic aspects of the workflow participant definition and are used in the sense of abstract actors. They are determined and assigned to concrete human(s) and systems during a workflow execution.

The dynamic elements of the above WPA definition are limited and do not meet the defined requirements. It is very hard to express dynamic elements of the definition that rely on control and audit data as well as relevant data that needs to be evaluated (such as relations on organisational model, user information, etc.).

Moreover, the WfMC description of the Interface 1 only suggests what can be done if more than one participant is assigned to a given activity. The description does not cover situations when selection of participants who will perform a given activity (from a set of assigned participants) should be done manually (ad-hoc). Such situation can be vital for workflows having participant assignments not well defined, thus dependent on human decisions during workflow executions. In addition, the WfMC's definition of WPA is not prepared to introduce the reuse mechanism, which should make it possible to build a WPA on the basis of other defined WPAs.

2.3 A New Approach Based on WPA and Expressed in WPAL

An approach that can express the above dynamic elements and satisfy the requirements for a modern WFMS is based on an extension of the WPA definition. In such an approach, a WPA is a *function* that specifies (explicitly or implicitly) a set of workflow participants that will perform a given activity. A WPA is based on set manipulation and can consist of:

- **Set of workflow participants** (in particular one participant) – an explicit enumeration of workflow participants - as in the WfMC definition,
- **Set operators** (union, intersection, and subtraction) that operate on sets of workflow participants, functions and other WPAs. The operators simplify the notation for complex WPAs,
- **WPA function** – evaluates a set of workflow participants on the basis of control, audit and relevant data. Functions are used to express complex relationships that cannot be represented by other elements of the WPA definition. Function arguments are WPAs too. A function can be implemented as a database procedure, a programming function (e.g. in C) or a class method (e.g. in Java). WPA functions can be reused in definitions of many WPAs,
- **Another WPA** – an already defined WPA. Instead of defining a new WPA from scratch it is possible to use already defined WPAs. For example, a set of employees that know Java and XML languages can be defined as an intersection of two separate WPAs – one that expresses the sets of employees that know Java and one that expresses employees that know XML. Using other WPAs, it is possible both to simplify the WPA definitions as well as to reduce the cost of the WPA developing and maintenance (e.g. the cost of testing),

The set of workflow participants that are the result of a WPA evaluation is called a *WPA evaluation set*. A WPA is equipped with two additional features: a *modifier* and a *decision*. They are used to delimit the WPA evaluation set that is the result of a given WPA evaluation. A modifier, on the basis of the set of participants evaluated by a WPA, determines the number of participants that will perform a given activity. The modifier can take one from two values:

- **One** – the activity is assigned to the first accepting participant;
- **All** - the activity is assigned to all the participants returned by a WPA.

A decision, on the basis of the set of participants evaluated by WPA, determines whether the set of participants will be assigned automatically or manually. It can take one from two values:

- **Auto** – WFMS automatically evaluates appropriate WPA and assigns the participants to a given activity;
- **Ad-hoc** – As previously, but the final assignment of the returned participants to a given activity is done manually by the performer of the current activity.

To prevent errors in a WPA definition, additional features of the WPA have been introduced: *cardinality* and *default WPA*. They are also present in other WFMSs, e.g. IBM MQSeries Workflow. The cardinality describes how many participants should be included into a WPA evaluation set. A default WPA describes participants that will be assigned to the activity if the real cardinality of a given WPA evaluation set is different from the defined cardinality (e.g. if a WPA evaluation set is empty). Default WPA should be defined explicitly as a set of workflow participants.

2.4 Representing User Groups in WPA

According to the WfMC definitions of a workflow participant and WPA it is possible to join users into logical groups. Groups based on an organisation model can be represented as organisational units. Other groups of users (e.g. related to functional structures) are represented by the role mechanism. If a user belongs to a given group, it has a role that represents the group (e.g. the role has the same name as the group).

Except organisational units, the definitions cause difficulties when one wants to determine a group through other defined groups. In the new WPA definition, a group can also be represented as a WPA. If a group is defined on basis of other groups, it is represented as a WPA that can be defined through other WPAs. For example, assume a WPA should represent users who know English and Polish, or being the president of an organisation. The WPA can be expressed as an intersection of two WPAs: people that know Polish and English, plus a one-element set of workflow participants – the president. For example, the defined WPAs would be as follows:

```
WPAExpEn = [Emp1,Emp2]
WPAExpPl = [Emp1,Emp3]
WPAExpEnPlPres = (WPAExpEn * WPAExpPl) + [President] =
[Emp1,President]
```

2.5 WPAL - a Language to Define WPAs

On the basis of the new definition of WPA a *Workflow Participant Assignment Language* (WPAL) has been created. Its syntax written in BNF is as follows:

```
<wpa_def> ::= <wpa_name>'='<wpa>
<wpa>     ::= <wpa1>{<op1><wpa>}
<op1>     ::= '+'|'-'
<wpa1>    ::= <wpa2>{<op2><wpa2>}
<op2>     ::= '*'
<wpa2>    ::= '('<wpa>')'|<set>|<function>|<wpa_name>
<set>     ::= '['<participant>{','<participant>}']'
<function>::= <function_name>'('<arg>{','<arg>}')'
<arg>     ::= <wpa_name>|<function>|<set>|<text>|<role>
```

Symbols used:

- '+' - a union operator,
- '*' - an intersection operator,
- '-' - a subtraction operator,
- `<wpa_name>` - the name of a WPA,
- `<function_name>` - the name of a function that is already defined,
- `<role>` - the name of a defined role,
- `<text>` - a string representing some description of WPA.

Some examples of a WPA in WPAL:

- A person who is an expert in Java and XML:
  `WpaA =Expert('JAVA') * Expert('XML')`

- A person, who is an expert in Java or Visual Basic, and knows XML (except Doe):
  `WpaB=(Expert('JAVA')+Expert('VB'))*Knows('XML')-['Doe']`
- Employees that earn less than 20000EUR/year and know Java and XML:
  `WpaC = Earn('LESS','EUR','YEAR',20000) * WpaA`
- A person who did the previous activity or the person who started the workflow:

```
WpaD = Participant(Prev_Activity) +
       Participant(Start_Activity)
```

2.6 Advantages of WPAL

The main extension of the new approach to existing ones is the definition of a WPA as a function that evaluates a set of workflow participants. It is possible to simplify the notation of many complex WPA definitions using classical set operators. Introducing WPA functions enables expressing complex relationships on control, audit and relevant data. The defined WPA functions can be used in many WPAs and the set of available functions can be easily extended. WPA functions can be grouped into sets of specialised functions, for example, a set of functions that operate on control data on order to evaluate a minimal set of workflow participants that have the lightest workload. New definition does not replace but extends the WfMC definition.

Since a WPA can include dynamic elements, it is not possible to verify it statically (like in WfMC). However, all merits of workflow simulation and testing can be applied to WPAL.

3 Practical Applications

WPAL is a part of Rodan Systems' knowledge and information management product family known as *OfficeObjects*®. Rodan Systems deployed *OfficeObjects*® at major Polish public institutions such as Ministry of Infrastructure, Ministry of Labour and Social Policy as well as commercial organisations such as Polish Press Agency and Authors' Association ZAiKS. At the first stage WPAL has been implemented in a workflow module that was built into *OfficeObjects*® *DocMan* – a document management system. The workflow process model has been based on the WfMC's process meta-model and the Process Definition Language (PDL). WPA functions were implemented as database stored procedures and able to express:

- participants that have already performed activities of a given workflow instance,
- relationships on organisational structure, such as:
  - the supervisor of a given workflow participant,
  - participants that are employed at a given position in a given organisational unit,
  - the organisational unit which the performer of a given activity belongs to;
- user groups that are created dynamically and may consist of other groups, such as:
  - a seller that belongs to the same region as a given customer,
  - dealers co-operating with a given company for more than two years;
- level of workload, such as performers who have the lightest workload.

A WPA could be made more precise through the modifier and decision attributes. For example, they helped to implement business processes allowing manual selection of workflow participants. The mechanism of checking WPA cardinality demonstrated its usefulness in serving cases where WPA returned the empty set.

The extended system has been deployed at several selected customers. After one-year collecting of the implementation results and new requirements they have been analysed from the point of the further workflow development view. Main results of the new workflow mechanisms were the following:

- Reducing the time of sending documents between workflow participants and the number of wrong participant assignments,
- Preserving acceptable system response time (from the users point of view),
- Confirming that successful implementation of workflow mechanisms is depended on the ability of a WFMS to integrate workflows with external applications and systems that already exist in an organisation,
- Demonstrating usefulness of WPAL, especially during the maintenance phase.

On the basis of the deployment results we have implemented a workflow management system – *OfficeObjects® WorkFlow*. It is a Java workflow component that can be integrated with practically any information management system. The *OO WorkFlow* architecture is consistent with the WfMC reference model and the process model is based on the WfMC process meta-model. Its structure is presented in Fig.1.

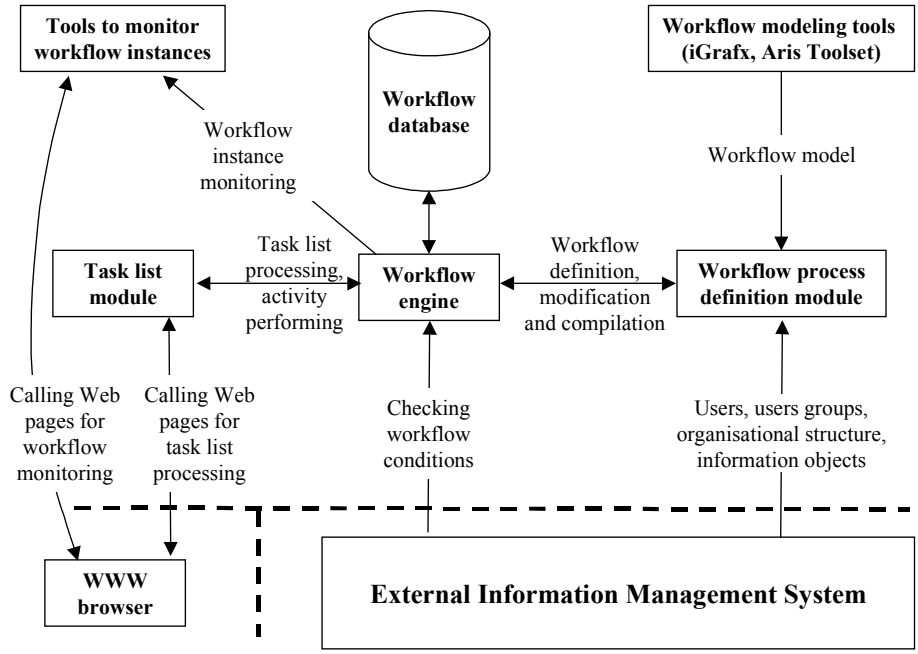

Fig. 1. The *OO WorkFlow* architecture

The OO Workflow consists of four basic modules:

- **Workflow process definition module** (WPD module), to define and modify workflow processes. A workflow definition is created in a graphic tool for modelling business processes such as iGrafx or Aris Toolset. The definition is then imported into the WPD module. On the basis of the business process model an adequate workflow definition is created. In order to achieve a complete workflow definition some additional information is to be added. For example, since dynamic aspects of WPA can not be expressed in the mentioned tools for business process modelling, they should be defined using WPAL in the WPD module. A complete workflow definition is saved into workflow database. WPAs are compiled into database stored procedures.
- **Workflow engine** – the main module that provides the run time execution environment for a workflow instances. In order to increase performance it is possible to install several workflow engines on different application servers.
- **Tools for monitoring workflow instances**. Such tools support other modules in information on workflow instances, namely history of their execution, possible bottleneck or deadlocks, etc.
- **Task list module**. This is a user interface supported with functions to process tasks (activities) assigned to a given user. In addition the module is able to execute third-party applications.

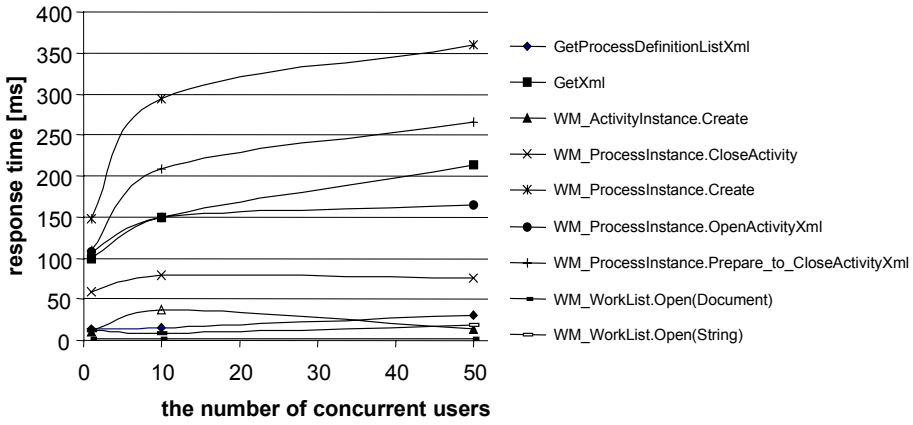

Fig. 2. The response time of the major functions of OO WorkFlow

A workflow database is any RDBM system that has a JDBC/ODBC connection. *OO WORKFLOW* has an object-oriented API. In order to assure flexibility of the component, data exchanged with the environment through API are defined in XML. During the heavy-load tests *OO WORKFLOW* showed that can be sufficiently productive WFMS. For example, we have tested the component for the following environment: about 200 000 workflow instances, and up to 50 concurrent users, Oracle database (8.1.5); Sun Solaris 5.8; Sun E450, one Ultra Sparc3, 433 MHz processor, used as a test server (i.e. with other systems tested in parallel).

The response time of major API methods was below one second. Moreover, along with growing the number of concurrent users, the response time increases sub-linear. Evaluation of WPAs in Prepare_to_CloseActivityXml method did not increase the response time.

Conclusions and Future Work

In this paper we have presented a part of an *OO WorkFlow*, a module for workflow management being a part of an entire system *OfficeObjects*®. The part is dealing with workflow participant assignment (WPA), a critical issue for dynamically changing workflows. WPAs are determined through simple yet very useful language WPAL. Simplicity and flexibility of such a language is critical because business processes are defined by casual users who hardly accept complex languages. WPAL is based on parameterised functions that can be defined in higher-level languages programming such as Java, and on intuitive set operators.

OO WorkFlow is currently a part of *OfficeObjects® Portal* that has been deployed at more than ten major polish public institutions as well as at commercial organisations. One-year experience with *OO WorkFlow* has confirmed usefulness of WPAL. Using WPAL the cost of implementing, testing, and - what is most important - maintaining workflow processes have been radically decreased.

Rodan Systems provides further development of WPAL in *OO WorkFlow*. Main features that will be extended in WPAL are the following:

- Automatic optimisation of WPA definitions. In many cases WPAs can be simplified to reduce the run-time evaluation overhead,
- Creating a library of functions available in WPAL. Implementations of *OO WorkFlow* have showed that there is a need to group functions that concern a particular issue/domain (e.g. work balancing functions, functions that are connected with organisation structure, and so on),
- Better mechanisms for exception handling. With the current mechanism it is difficult to detect such situations as:
  - a WPA evaluation set includes a workflow participant that has no rights to perform a given activity;
  - a workflow participant in a given WPA is included more than one time.

There are future plans concerning the application of WPA mechanism to evaluate the set of users that can perform a given function/service on information objects in *OfficeObjects® Portal*, a system for making large Web object-oriented applications.

References

1. W.M.P. van der Aalst. *Flexible workflow management systems: An approach based on generic process models*. Proc. of the 10th International Conference on Database and Expert Systems Applications (DEXA'99), volume 1677 of Lecture Notes in Computer Science, pages 186-195. Springer-Verlag, Berlin, 1999.
2. W.M.P. van der Aalst, T. Basten, H.W.M. Verbeek, P.A.C. Verkoulen and M. Voorhoeve. *Adaptive Workflow: On the interplay between flexibility and support*. Proc. of the first International Conference on Enterprise Information Systems, Setubal, Portugal, pages 353-360, 1998
3. C.A. Ellis. K. Keddara, and G. Rozenberg. *Dynamic change within workflow systems*. In Proc. of ACM Conference on Organisational Computing Systems (COOCS 95)
4. C.A. Ellis. K. Keddara, and J. Wainer. *Modelling workflow dynamic changes using time hybrid flow*. In W.M.P. van der Aalst, G. De Michelis, and C.A. Ellis, editors, Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98), volume 98/7 of Computing Science Reports, pages 109-128. Eindhoven University of Technology, Eindhoven, 1998.
5. Y. Han and A. Sheth. *On adaptive workflow modelling*. Proc. of the 4th international Conference on Information System Analysis and Synthesis, pages 108-116, Orlando, Florida, July 1998
6. J. Klingemann. *Controlled flexibility in workflow management*. Proc. of the 12th International Conference on Advanced Information Systems Engineering (CaiSE'00), Stockholm, Sweden, pages 126-141, 2000
7. M. Momotko. *Implementation of workflow processes in OfficeObjects DocMan*. Proc. of the 2nd Polish Conference on Software Engineering, Zakopane, Poland, Oct. 2000. In Polish.
8. M. Momotko. *OfficeObjects WorkFlow – an independent workflow management component*. Proc. of the 3rd Polish Conference on Software Engineering, Otwock, Poland, Oct. 2001. In Polish.
9. M. Reichert and P. Dadam. *ADAPTflex: Supporting dynamic changes of workflow without loosing control*. Journal of Intelligent Information Systems, 10(2):93-129, 1998.
10. A. Sheth. *From contemporary workflow process automation to dynamic work activity coordination and collaboration*. Siggroup Bulletin, 18(3):17-20, 1997
11. M. Voorhoeve and W.M.P. var den Aalst. *Conservative adaptation of workflow*. Proc. of the International Conference on Practical Aspects of Knowledge Management (PAKM'96), Workshop on Adaptive Workflow, pages 1-15, Basel, Switzerland, 1996
12. M. Voorhoeve and W.M.P. var den Aalst. *Ad-hoc Workflow: problems and solutions.* Proc. of the 8th DEXA Conference on Database and Expert Systems Applications, pages 36-41, Toulouse, France, 1997
13. WfMC, *WorkFlow Process Definition Interface – XML Process Definition Language*; WfMC-TC-1025 (Draft 0.03a); May, 22, 2001
14. WfMC, *WorkFlow Management Coalition – Terminology & Glossary*; WfMC-TC-1011 (Issue 3.0); Feb, 1999
15. WfMC, *Workflow handbook*. L. Fisher, ed., Future Strategies Inc. 2001