

Workey: a method and tool to build web-enabled workflows

Jean-Loup Chappelet, André Le Grand

Institut de Hautes Etudes en Administration Publique
21 route de la Maladière, CH-1022 Chavannes-près-Renens, Switzerland
{Jean-Loup.Chappelet,Andre.LeGrand}@idheap.unil.ch

Abstract. Workflow management systems provide functions for process modeling, implementation and automation. Many systems do not allow modeling organizational goals and relating these goals with the processes which enable them. The method OSSAD provides models and project management guidelines to this effort. But this method is not sufficient to fully support workflow management. In order to do so, we define Workey as an extension of OSSAD, to enable workflow specification, implementation and automation for the web.

1 Introduction

Workflow management systems generally provide functions for process modeling, implementation and automation. The management of workflows involves [1]:

- *process modeling and workflow specification.*
This phase requires models of workflow and methods to represent a process as a workflow specification,
- *process reengineering.*
This phase requires methods for the optimization of processes and,
- *implementation and process automation.*
This phase requires methods and technologies to coordinate systems and users to implement, plan, carry out and control the tasks of workflow such as they were specified.

The next subsections of this introduction describe the limits of current solutions according to the above topics. Then we discuss briefly the qualities and drawbacks of Internet for workflow applications. Thereafter we present the strong points of our contribution.

1.1 Process modeling

Following a recent workshop on workflows and process automation in information systems [2], some participants wrote a report describing the state of the art and the prospects for the field which they named "work activity coordination" [2]. This report enumerates the methodological points for which it is necessary to define a more rigorous framework, among those:

- the representation of process definitions,
- the modeling of coordination and control,
- the understanding of which methods and tools apply in which situation,
- and the consistent use of modeling concepts.

The formalism and the project management guidelines of OSSAD¹ [3] provide an interesting framework to address the above points [4][5]². This method should however be extended to enable workflow specification and automation. The extensions of [4] aim to model the actors' responsibilities related to a workflow, and the circulation of documents. The models introduced in this extension do not provide a sufficiently formal specification from which a tool can automatically build a workflow application.

There are two paradigms of process modeling[6]: one based on the communication between people, the other based on the activities. The approach based on the communication comes from Workflow Action[7]. Its designers consider that the objective of reengineering is to improve customer satisfaction. Thus, every action is modeled as a loop of four phases of communication between a customer and a performer: preparation, negotiation, performance and acceptance [7]. Action Workflow is not always well adapted to the modeling of administrative workflows: many procedures cannot easily be modeled as a network of nested loops of customer - performer.

Activity-based methods aim at modeling the tasks and their ordering rather than the agreements between human actors in their communications. Most of workflow management systems are activity-based. Workey is also based on activities. Workey extends OSSAD to transform descriptions of procedures into specifications from which workflow applications on Internet are automatically implemented. An analysis is usually performed before the implementation of these specifications. The analysis results may lead to reengineering the processes.

1.2 Process reengineering

Process reengineering goal is to optimize processes with respect to criteria like customer satisfaction, cost reduction, or the introduction of new services. This reengineering is sometimes regarded as an art [1]. However, in order to imagine new solutions, methods and tools for process analysis -by providing qualitative and quantitative information on the execution or the runnability of the specifications- guide and stimulate creative thinking. According to [1], the capabilities for analysis, testing and debugging of specifications are insufficient in the current systems. They allow, for example, the animation of processes or simulation. These tests are carried out without requiring the effective implementation of the application to analyze. Additional tests can be carried out on the application when it is implemented. Pantha Rei [8], approach based on an extension

¹ Office Systems Support Analysis and Design. For a more complete list of reference about OSSAD, please see <http://www.unil.ch/idheap/ossad/ossad>.

² Workey is an extension of motown for web applications

of the PERT method, tries for instance, to anticipate the delays during the execution of processes. Even this approach is inapplicable if a process contains a loop, this system analyzes processes during their run, whereas others approaches analyze them before their implementation.

An OSSAD project incorporates a phase for the design of alternative solutions. Its purpose is to define other ways of realization of processes. These types of implementation of processes are called procedures in OSSAD. The confrontation of these alternative solutions is carried out by the project team on the basis of its experience (speculative mode). Workey supplements this mode with tools for qualitative analysis and simulations of time and cost [5].

These tools help for instance, to identify bottlenecks. The relevance of any simulation requires a rigorous and accurate quantification of the parameters it uses. This quantification involves considerable work. However, concrete values, extracted from the history of the real use of an application can be provided as input for simulation. Such simulations, even if they require a great effort of quantification, provide relevant information for process reengineering.

The results of analysis having guided the choice among these various alternatives of design, the implementation of an application can be performed. The application can in its turn provide real data for analysis and simulation.

1.3 Implementation and automation

Process implementation The implementation can be carried out, either by a team of developers starting from the specifications, or, and it is the more frequent case, by automatic generation of rules which are interpreted by the workflow management system.

In order to implement workflow applications, there are some myths about using Internet which should be carefully studied:

- the concept of "universal client" hides many and important disparities between the various browsers and their successive versions.
- there are several configurations for a single browser. Thus the storage and retrieval information on the client side of the connection ("cookies") can be enabled /disabled. An organization may impose a policy which cannot be changed for various reasons. The browser configuration may also prevent the handling (by script) of the history of previously accessed pages (in order to erase the already submitted pages).

Ultimately, the concept of "universal client" of a workflow management system for Internet is reduced to a small set of functions.

Process automation It concerns the planning and the control of the process executions. The automation of the processes is carried out according to the *push* or *pull* mode. In the push mode, the workflow management system dispatches the documents to users according to their workload. In the pull mode, the users themselves will seek the documents on which they have to work. In Workey,

the default mode is pull. However, for particular operations, if specified in the process definition, Workey pushes documents towards one or more users (see concepts of selection and notification in section 2.3.).

For the workflow management system (in push or pull mode), using Internet as a client/server platform has undeniable qualities:

- easy deployment of the workflow applications within the organization,
- reduced training of the users (for the basic functions of the browsers),
- easier access to external clients,
- mobile computing, with machine and operating systems independence,
- inter-organizational workflows,

Using browsers as clients, however imposes more constraints than a proprietary client/server solution for which client-server relations can be tailored. Indeed the problems of security are then increased. Moreover the HTTP protocol which is state-less, limits the tailoring of these relations.

Inherent and specific to the use of browsers, the problems concerning the execution control of the workflows are primarily related to the actions "back" and "forward". These two actions give access to previously accessed pages in the same session of work. These pages display workflow documents which may have changed since they have been loaded. Therefore the double submission of the pages and loss of time due to a work performed twice by a user are to be avoided. It is then necessary, either to force the reloading of these documents by using HTML meta-tags (client-side solution), or to prevent double submission (server-side solution). These problems do not arise in proprietary solutions because the developers have the control of the dialogues between the clients and the server. Evolving from proprietary systems to Internet leads to the loss of control over the client configuration.

The next section draws the strong points of our contribution on the methodological and technical aspects.

1.4 Strong points of our contribution

In spite of the efforts of WfMC³, a rigorous conceptual framework is still missing for the modeling of workflows. OSSAD brings conceptual and methodological elements which are very useful and pertinent for organization modeling. Workey extends OSSAD to model, implement and automate workflows. Workey adds:

- a prescriptive level to OSSAD to specify workflow applications,
- a step by step guide to build specifications,
- time and cost simulation functions for the analysis of these specifications,
- an application builder for web-enabled workflow applications.

The rest of this article is organized as follows. Section 2 describes the modeling of the processes with Workey. This section introduces the various levels of modeling of OSSAD and presents the extensions for the modeling of workflows.

³ workflow management coalition (<http://www.aiim.org/wfmc>)

The implementation and the architecture of the workflow management system are described in section 3. Section 4 describes the process automation. In these two sections, the use of Internet is compared with a proprietary solution. In conclusion, we present some research perspectives resulting from this work.

2 Process modeling

Processes are modeled at three levels: abstract, descriptive and prescriptive.

2.1 The abstract level

It describes the objectives, goals or missions of an organization, without taking into account the human and technical means necessary for their realization. These abstract models are generally invariants of a means reconception ("business process *redesign*"), but not of a major recasting of the activities of the organization ("business process *reengineering*"). The abstract level has a sole model whose components are *functions*, *sub-functions*, *packets* and *activities*.

A function is a subset of the organization with homogeneous objectives. Functions exchange data information, named packets (e.g. fig 1). Packets are not necessarily information related to the goals of the function. It is not the aim of finance and accounting to produce requirements, whereas the application development is to produce software products to the demanding function. Functions are not departments of a company. They describe a goal this enterprise has to achieve.

The functions can be broken down into sub-functions. The unbroken functions or sub-functions are called activities. These activities constitute the finest level of detail of the modeling of the objectives.

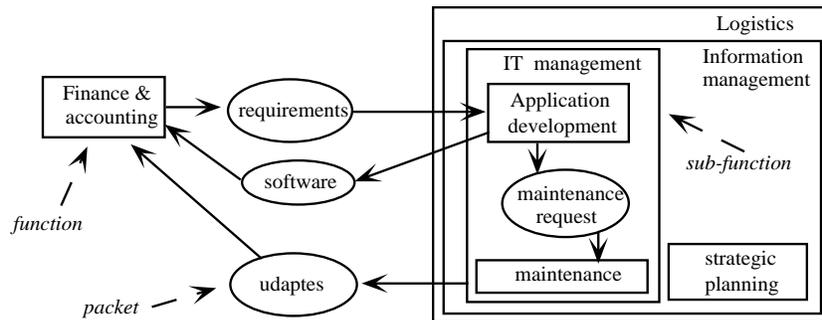


Fig. 1. abstract level: functions decomposition and packets

2.2 The descriptive level

The models of this level concern the human, technical and organizational support systems to achieve the objectives detailed in the form of activities. An activity in the abstract level maps onto a *procedure* in the descriptive level. A procedure is a coherent set of *operations*; it is carried out cooperatively by a set of *actors* to which *roles* are assigned. An actor may carry out several roles, a role may be carried out by several actors. A packet in the abstract level maps onto a set of information *resources* in the descriptive level. This level contains three models:

- the *procedure model* which describes the exchange of information between the procedures,
- the *role model* which describes the exchange of information between the roles and,
- the *operation model* which describes the flow of operations of a procedure.

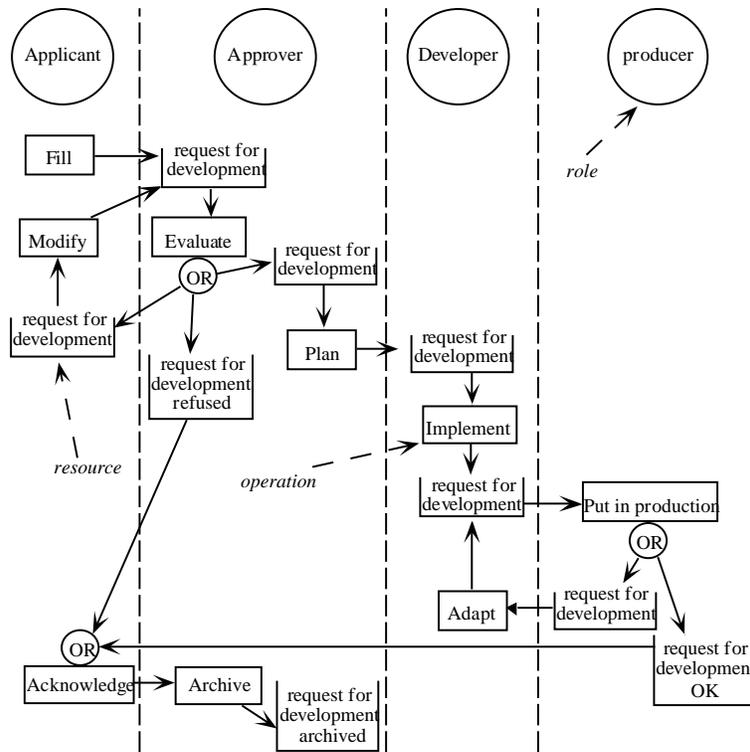


Fig. 2. descriptive level: operation model of a procedure

For example, fig. 2 displays the operation model of the procedure "application development". This procedure directly corresponds to the activity with the

same name into fig. 1. Four roles are involved : Applicant, Approver, Developer and Producer. The *task* of a role is the set of operations which are placed in the role's column.

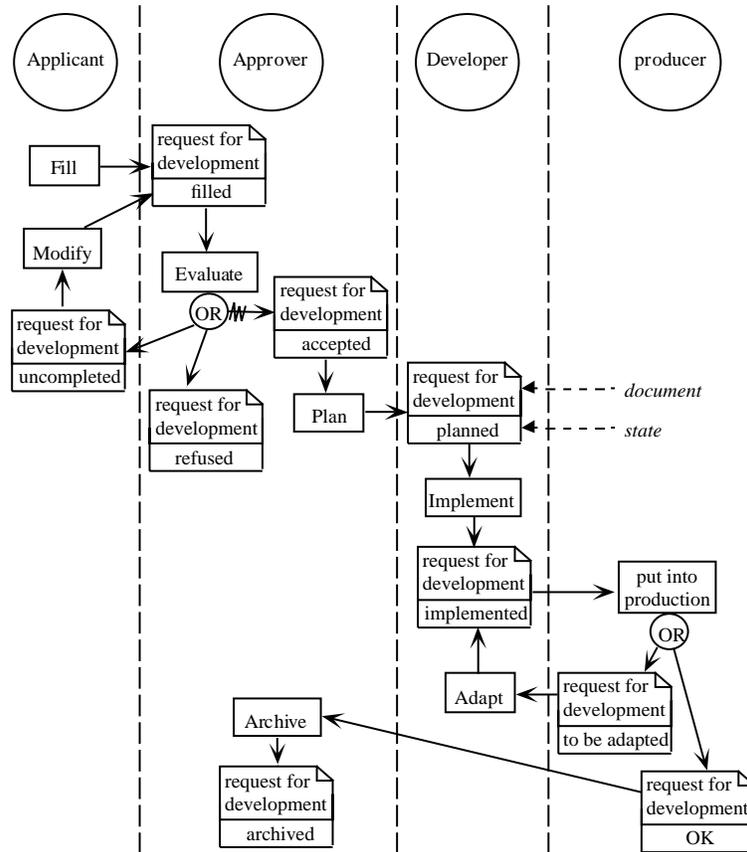


Fig. 3. prescriptive level: a workflow model

2.3 The prescriptive level

It extends the operation model by the specification of what will be automated in a workflow. To this end we introduced the following concepts:

- document,
- state of document,
- structure of document
- constraint of prohibition and obligation,

- selection, and,
- notification.

A *document* is a resource of the operations model which is computerized. The state of a document is used to indicate the document status within a flow. The *structure* of a document is a set of informational fields. *Constraints* between two operations can be defined: they oblige / prohibit a same actor carrying out these two operations on the same document.

For instance (e.g. fig. 3), a prohibition constraint between the *Fill* and *Evaluate* operations will prevent the same actor to fill and positively evaluate his own requests, although he can still plan them.

The concept of *selection* is to choose a subset of the actors allowed to carry out the next operation.

For example (e.g. fig. 3) when an actor of the *approver* role has *accepted a request for development*, he can choose a subset of actors of the developer role who are allowed to *plan* this request. For the other outputs of the *evaluate* operation, there is no selection.

The concept of *notification* is related to automation of a procedure. It indicates that the actors of a role must be notified by an electronic message when a document reaches a particular state and that the next operation belongs to the task of this role.

Fig. 3 displays the workflow model of fig. 2.

2.4 Steps of specification

The transformation of a descriptive model into a prescriptive model is done according to the following steps:

1. identify the information resources which will be computer supported,
These resources become documents. When resources become documents, some "read" operations can be deleted since Workey lets users see the state of their documents within the flow, for instance, *acknowledge* on fig. 2.
2. specify the states changes of these documents (including computed changes),
3. specify the constraints between operations, if necessary,
4. determine the states for which it is necessary to select the actor or the actors having to carry out the following operation. This selection can be associated with a notification by an electronic message.
The notifications are recommended either for occasional users of an application or for users working on several applications of workflow.
5. specify the structure (fields and sections) of the documents.

The following paragraph describes the transformation of these prescriptive models into an application.

3 Process implementation

The Workey application builder makes a workflow application based on Lotus Notes/ Domino. This application is a standard Notes base of documents to which we added a workflow engine that we developed. This target system provides a high level of security for access controls. The architecture which mixes Notes and Workey, is described in the next subsection.

3.1 Structure of Workey applications

Domino is a HTTP server which generates dynamically HTML pages from the documents stored in Notes bases. These bases are then accessible via browsers, while preserving their access rights. Domino and Workey are integrated in the

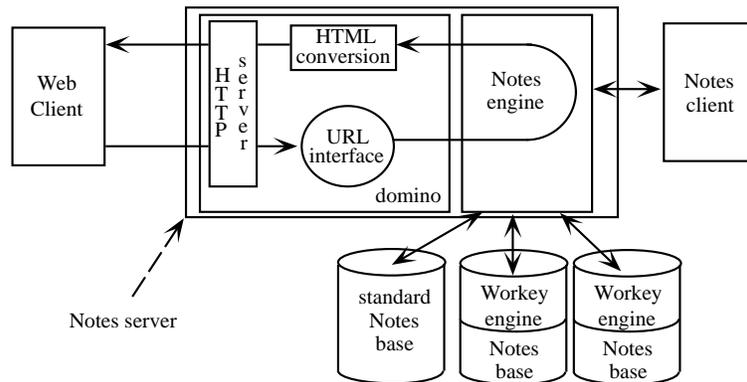


Fig. 4. Architecture of Workey applications with Lotus Domino

following way (fig. 4):

- the server is a Notes/Domino server.
- the clients are Web browsers or Notes clients,
- the workflow bases include the Workey workflow engine.

The same application can be accessed by a Notes client or a web browser. The users, whatever their client type is, can only access the bases for which they have rights.

The implementation does not amount to only adding a workflow engine to a Notes base. The following section presents the details of implementation.

3.2 Implementation of a workflow application

Fig. 5 describes the implementation procedure. It has two roles: designer and builder. Workey Builder carries out the builder role and generates a Notes base.

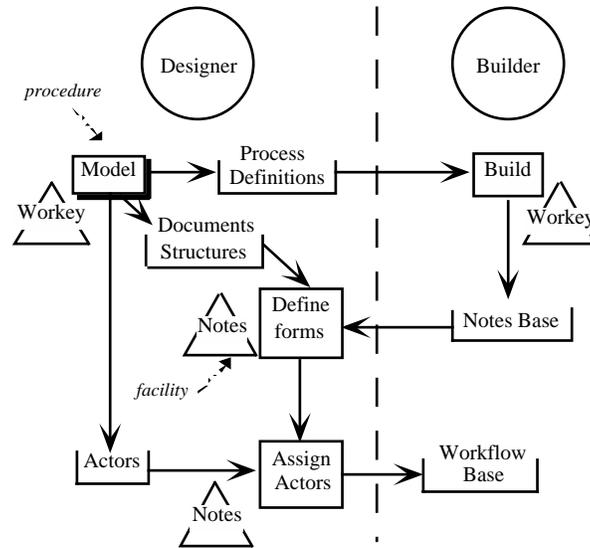


Fig. 5. implementation of a specification

This Notes base contains an workflow engine and the rules derived from the procedures specifications. The following section describes how the actors perform their work with Workey.

4 Process automation

The workflow engine works differently, depending on the type of client, since the HTTP protocol is state-less. The Notes/Domino server which recognizes the client type, controls the workflow engine in relation with its type. With a browser, the workflow engine is requested at each edition and recording of a document. When a document is queried for edition, the workflow engine adds a section to push or not the document one step further within a flow (e.g. fig. 6).

The "Workflow status" column is to specify whether the document moves or not within the flow. "Unchanged state" is to tell Workey that the modifications made during the edition do not lead to a new state (the current operation is not fully achieved). The other choices are to indicate the possible new states of the document. The "Next actors" column may enable a selection of the actors carrying out the role of the next operation. If there is no such specification at the prescriptive level, all the actors of this role are allowed to perform the next operation.

For a request of development in the *filled* state, the *Evaluate* operation leads possibly to various new states: *uncompleted*, *refused* or *accepted*. The actor has to tell Workey the effects of his changes on the document being edited, and then he has to push the *submit* button, to record the document. For the

refused and *uncompleted* states which do not allow any selection, this column is empty. On fig. 6, since the *accepted* state is chosen, the list of actors of the *plan* operation is displayed for selection. The list of actors also depends on the constraints between operations, if any.

The screenshot shows a web-based interface for document management. At the top, it displays the document title 'André Le Grand / mail:agent to delete old mails'. Below this, the 'Current state' is 'filled' and the 'Next operation' is 'Evaluate'. A sidebar on the left contains navigation options: 'Process & graph', 'Operations', 'Act', and 'Document - state'. The main content area is divided into two sections: '1) Workflow status' and '2) Next actors'. Under '1) Workflow status', there are three radio button options: 'Unchanged state', 'Request for development/refused', and 'Request for development/accepted' (which is selected). Under '2) Next actors', there are two radio button options: 'without restriction' (selected) and 'restrict to'. A list box on the right shows the names of potential actors: John Smith, Michael Ford, and Evelyn J. Nickolson. At the bottom, there is a 'Contextual help' link and a 'Submit' button.

Fig. 6. selection of the next actors in relation with a new state

When the user submits the document, Domino requests the workflow engine to compute the document's update. If its state remains unchanged, the document is only recorded, if not, the document is updated according to its new state and the workflow rules.

The differences of interactions between the workflow engine and the client, according to its type, are summarized below:

- with a Notes client, the workflow engine only interacts with the client whenever an information about a document circulation is required [5].

For instance, for a document in *filled* state, the workflow engine asks the actor to choose the new state. If the actor chooses *accepted* as new state of this document, since there is a selection specified (see fig. 3), then the workflow engine will transfer to the client the list of actors of the next operation. The actor, will then be able to select the next ones.

- with a browser, the workflow engine transfers to the client all the information. Then all the dialogues to specify the document circulation are made on the client-side.

The Domino servers do not have yet all the capabilities of the Notes servers. The administration of the Workey applications is thus carried out from a Notes Client software, like for instance, the policy of filing documents.

Moreover the roles and users administration, the evolution of the document forms and the management of evolving specifications are also performed inside Notes. These three functions cover the various changes which the modeling of a process can undergo. The management of this evolution is detailed in [5].

5 Conclusion

Our work consisted of extending an office work modeling method by the addition of workflow management capabilities. This extension consists in the definition of a third level of modeling to the OSSAD method, namely the prescriptive level. These ideas are implemented in a CABRE (Computer-Aided Business Re-Engineering) software which enables:

- modeling processes and procedures according to OSSAD and its prescriptive extension which is centered on the concept of document,
- carrying out qualitative and quantitative analysis of models,
- building web-enabled workflow applications from the procedures specifications. These applications are based on Lotus Notes/Domino and the Workey workflow engine.

This work revealed some perspectives which exceed the framework of our approach. The canceling of a procedure after it started is still in our view a research issue. On one hand, some models and tools to analyze how a procedure can be canceled are necessary to understand the cancellation effects, and on the other hand, workflow management systems should include functions to carry out properly these cancellations. Another interesting perspective concerns the management of process evolution: a method should provide guidelines to develop alternative designs in relation with the objectives of the evolution and the results of simulations.

References

1. Georgakopolous D., Hornik M., Sheth A.: An overview of workflow management: from process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3 (1995), 119-153.
2. NSF Workshop on Workflow and Process Automation in Information Systems: State-of- the-art and Future Directions, http://lstdis.cs.uga.edu/activities/NSF-workflow/proc_cover.html (1996)
3. Conrath D.W., Dumas P. et al.: *Office Support Systems Analysis and Design: a Manual*. Esprit Project 285. IOT, Munchen (1989).
4. Nurcan S. "Analyse et conception de systèmes d'information coopératifs. *Technique et Science Informatique*, vol. 15(9),(1996), 1287-1315.
5. Chappelet J-L., Le Grand A., Prevel M., Snella J-J.: Motown: a practical approach to workflows. proceedings of the 1st east-european symposium on Advances in Databases and Information Systems (ADBIS'97), St-Petersburg, Russia (1997).
6. Marshak R.: Software to support BPR- the value of capturing process definitions, *Workgroup Computing Report*, Patricia Seybold Group, vol. 17, no. 7 (1994).
7. Médina-Mora R., Winograd T., Flores R. et Flores F.: The action workflow approach to workflow management technology. Proceedings of the conference on Computer-Supported Cooperative Work, (1992).
8. Pozewaunig H., Eder J., Liebhart W.: ePert: extending pert for workflow management systems. Proceedings of the 1st east-european symposium on Advances in Databases and Information Systems (ADBIS'97),St-Petersburg, Russia (1997).