RZ 3116 (# 93162) 03/15/99
Computer Science/Mathematics 14 pages

Research Report

OpenWater White Paper

Keith Whittingham

IBM Research Division
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

IBM Research Division
Almaden · Austin · Beijing · Haifa · T.J. Watson · Tokyo · Zurich

OpenWater White Paper

Keith Whittingham

IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract

Because of internal factors like technology changes and external forces such as legislation and competition, business procedures must constantly adapt and evolve in order that the organization remains viable. In addition, the need to handle many exceptions makes business processes complex and modeling them in workflow management systems an extremely difficult task. We propose a workflow support system that, by enforcing business rules only where necessary, allows the users to solve day-to-day business problems in a pragmatic way. In addition, the knowledge gained by watching the participants of a business process can be shared across the enterprise with the goal of constantly improving organizational efficiency.

Keywords: CSCW, workflow, knowledge management, organizational learning, business process reengineering, organizational memory.

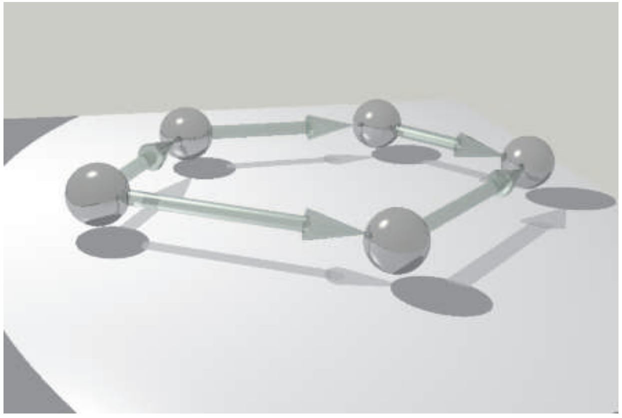

Figure 1: The ECF browser main window

1 Introduction

OpenWater is an ongoing research project at the IBM Zürich Research Laboratory in Switzerland that brings together workflow, knowledge management and organizational learning. The goal of the project is to use a collaborative work support tool to collect information about how business processes are really carried out in an office environment and to use this knowledge to streamline the organizational activities. This white paper contains a non-technical description that provides an introduction to the project.

The document is split up into the following sections:

- Background: this contains an overview of the technologies and research areas that play a part in OpenWater. Although these sections may be skipped if you are familiar with the subjects, be sure to read the WfMSs weaknesses and organizational learning sections as they describe in detail the problem areas that OpenWater addresses.
- The OpenWater system: a user's perspective of OpenWater; how it works and what it does.
- The pilot: a description of the pilot.
- Discussion: some observations and objective discussion of various important aspects of OpenWater.
- Conclusions: summing up and some speculation on what future work is likely to be.

After reading this document you should have a clear picture of how OpenWater works and how it helps to improve the efficiency of an organization.

2 Background

Two topics seem to be the subject of lengthy debate in both the literature and business press: workflow support and knowledge management. Another, as yet lesser known, subject that looks like having an impact on the business world in the future is organizational learning. At first sight these three disciplines seem to be distinct but we shall see that the three can exist together in a kind of symbiosis: the output from one benefiting the other and vice versa.

This section gives a brief overview of these three subjects in isolation. We can then go on in the following sections to explore how they can be combined.

2.1 Workflow management systems

Around the 1930's Alfred Sloan, founder of the MIT's Sloan School of Management, applied the thinking of the industrial revolution to management activity. In more recent years we have witnessed collaborative work software products like Lotus Notes, SAP's System R/3 line, BAAN and MQ Series Workflow [1] from IBM making a further step and applying production line thinking to business process management. Whilst there are undoubtedly benefits to this approach, the introduction of such systems into the office environment has not be an easy task. We shall describe workflow management systems (WfMSs) and see that most of the problems associated with them tend to be due to their inflexibility particularly when dealing with ad hoc processes and exceptions to normal business process handling.

A WfMSs is a computer system belonging to a group collectively known as computer supported collaborative work (CSCW) systems. CSCW systems have, and continue to, gain importance in the business world as intranets and Internet access become the norm. There are many different types of software which can be considered 'CSCW' for example Lotus Notes, Microsoft Exchange and of course electronic mail. Many of these claim to support workflow; WfMSs are unique in that they are very 'business process aware'.

We distinguish between the terms workflow **management** and workflow **support** in the following way. By workflow management we imply that the flow of work is pre-defined and computerized. A WfMSs, as we shall see, is an example of a system that manages workflow. In contrast by workflow support we imply that the flow of work is directed by the participants rather like email; OpenWater *supports* workflow in this way.

For the purposes of this document we define a WfMS as a system that conforms to the Workflow Management Coalition reference model [3]. It describes a number of components: server, client etc., that together support workflow. IBM's MQ Series Workflow is such a system. WfMSs distinguish themselves in the way that they abstract the flow of work between the participants of a business process from the actual performance of the activities which make up that business process.

Using a WfMS is a two stage process: build-time and run-time. During the build-time phase the business process must be understood and captured in a computerized model typically with the help of a graphical tool. The model, called a process template, must then be tested, debugged and simulated. The process template is ready for use. In the run-time phase users access the central system via a client interface. When a process starts in the 'real world,' say, an insurance claim is received, then an instance of the respective process template is started in the WfMS by a user via the client interface. Users involved in the process then automatically get notifications of tasks they are to perform in the form of work items in a to-do list.

We make the observation that the implementation of a business process in a WfMS is essentially the same as that of writing a traditional computer program. It might be possible to implement very simple processes quite quickly but this is the exception, not the rule. The consequences of this is that the modeling process requires similar effort and commitment by the organization to that of conventional software development.

We also comment that the modeling exercise is typically carried out by workflow specialists who acquire knowledge of the process to be automated by interviewing the people that have traditionally performed the process. Once the business process is implemented the experts typically move on to other projects.

Finally we stress that the users no longer have the opportunity to take decisions as to the performance of a particular process instance - the work flow is entirely controlled by the workflow system according to the definition of the process.

Because of the way in which they work, WfMSs are suitable for automation of business processes that are very well defined, are frequently executed and have few exceptions. In order to broaden their application, state of the art WfMSs are becoming more flexible. A significant effort is now being made in the research community [2] to pursue so-called adaptable workflow systems - OpenWater is just such a system.

We now look, in turn, at the benefits and weaknesses of WfMSs.

Benefits

The most advertised benefit of owning a WfMS is improved quality of service. The WfMS introduces an electronic representation of a process in the computer which prompts the participants of the workflow to perform their various tasks in the correct order and with a minimum of delay.

Secondly, because the WfMS controls the flow of work and oversees the process it can guarantee compliance to business and safety regulations. The participants in a business process and their respective responsibilities can be easily identified. Monitoring the business process and setting alarms for important tasks not completed on time becomes possible. The WfMS also archives the process performance providing a useful audit trail of process instances.

Weaknesses

The single biggest problem associated with WfMSs is that of accurately capturing the business process. This is analogous to getting ISO 9000 certification - business processes tend to be complex and have many exceptions and deviations from the norm. If a process is to be automated using a WfMS then the process and all its peculiarities must be accurately modeled. This requires a significant amount of time and expertise.

WfMSs are typically not very good at handling ad hoc workflows and unforeseen exceptions. Most WfMSs do not allow runtime modification of the workflow although systems such as PowerFlow and Keyfile do support intervention by the user using a graphical tool similar to that used in the build-time phase. At first sight this may seem to resolve the problem but it must be remembered that most people do not, and would not, want to understand the complexities of the workflow underpinning the business process they work on. This makes modification of a process instance's workflow very difficult in practice in all but very rare cases.

Lastly WfMSs tend to alienate the participants of a workflow from each other and from the process. For example: ask yourself what happens to a travel expense claim after you send it to your manager to sign? As participants, even in a traditional paper-driven system, we have a limited view of the process we are involved in: typically we understand only the next or the next two steps. One of the goals of WfMSs is to limit this view even further: a user is sent work items to perform and does them: the predecessors and successors in the process are hidden from the user. There are two downsides to this alienation. Firstly less job satisfaction - the cog-in-the-machine syndrome. Secondly, and perhaps more importantly, as the details of process are lost to the WfMS the users forfeit the ability to ask questions like "Does this make sense?" and "Wouldn't it be better if...?"

2.2 Knowledge management

As knowledge takes its place along with the traditional assets of labor, capital, land and raw materials as a recognized valuable asset we instinctively realize that it's a good thing to manage it. The subject of knowledge management is now gaining interest at senior executive levels as is evidenced by the setting up of bodies like IBM's Institute of Knowledge Management [6]. So just what is knowledge management? A brief check of the literature [7] yields many different answers, here's just a couple:

Information Week describes knowledge management as a four stage process; gathering (e.g. OCR), organizing (e.g. indexing), refining (e.g. mining) and disseminating (e.g. alerting). *16 March 1998, "Knowledge Management: Great Concept But What Is It?"*

The Gartner Group Inc, 1996: "...a discipline that promotes an integrated approach to identifying, managing and sharing all of an enterprise's information assets. These information assets may include databases, documents..."

We won't try and add to these definitions here but we will make the observation that we've been managing information for some time now so if we can define what knowledge is we should be closer to understanding knowledge management.

OpenWater's brand of knowledge management

As an illustration we'll take a look at an imaginary company: Barns & Pollack. They have a carpenter, Carl, who works in the office and makes furniture, desks, tables, shelves and the like. Carl is very good at his job but is getting on and is soon to retire. Carl's manager Martin, knowing that he is going to find it difficult to find a replacement with Carl's know-how, has the idea of getting Carl to write everything he knows about making office furniture. Shortly after Carl's departure Martin realizes that they need a new chair for some visitors. Martin hands Carl's pile of notes to his secretary: "Could you make me two chairs by Friday - everything you need to know should be here...."

Of course the situation is comical: the secretary will not be able to make the chairs; even though she may be in possession of the necessary *knowledge*, she simply doesn't have the skills. To get the chairs made in the time available Martin must hire another carpenter.

As that attempt at knowledge management doesn't seem to work let's assume that Martin hired a replacement for Carl and ask ourselves what would it have been useful for Carl to write down? Probably you'll come up with answers like where the furniture plans are stored, what materials to order, where to order materials from, how long they normally take to arrive, what paper work has to be filled in, what has to be sent to accounts, etc. That is the knowledge that surrounds the 'make-chair' activity, the knowledge of the process. We often call this knowledge 'experience', especially when a particular problem has been solved many times.

We are now in a position to better understand the difference between information and knowledge. We can think of information as a collection of facts: 'the order was sent to Woodways', 'the receipt was sent to accounting'. Knowledge can be thought of as place information in the context of experience: 'normally the materials arrive in about three days', 'it costs about $80.00 in raw materials to make a these chairs'.

What we have described is a record of the interaction between participants of a process; that is, the members of the organization and we class this it as organizational knowledge. Organizational knowledge is knowledge found not in an individual but in the organizational collective. It consists of values, action strategies, assumptions and expectations [5].

As we might expect the challenges inherent in achieving some form of knowledge management in an organization are:

- Assimilation: the collection of the raw data. Asking members of an organization to write down everything they know and everything they do is simply impractical
- Organization: structuring the data in an accessible way. We tend to store information in such a way that some questions can be answer easily - others cannot.
- Dissemination: getting the right information in the right form to the right person

Finally we have to ask ourselves why we should try to manage knowledge? Having knowledge littered around the office doesn't, on it's own, offer any tangible benefits. The knowledge has to be relevant and we have to use it in some way - organizational learning.

2.3 Organizational learning

We believe that by exploiting a particular type of knowledge - the knowledge of how a group of people perform a task, the organization can learn, and through this learning, benefit. The goal of knowledge management is organizational learning.

In 1978 Argyris and Schön [9] introduced the concept of organizational learning. Organizational learning refers to learning behavior within an organization, that is that the organization itself learns as an independent learning organism. Whilst that sounds like a very abstract idea what it really means is that the organizational knowledge can be improved and fine-tuned over time by organizational action.

Argyris and Schön go on to identify three types of learning called single-loop, double-loop and deutero-learning. Single loop learning is simply correcting errors: for example, if by accident 100 instead of 1000 widgets are shipped to a customer then shipping another 900 would resolve the problem. The problem is corrected but internal strategies and procedures do not change. In double-loop learning the strategies or values of the organization do change - here the organization adapts or evolves in response to some change in the environment - there's a lot of organizational double-loop learning going on at the moment in response to the emergence of the Internet. Finally, deutero-learning is defined as learning how to learn.

Making organizational learning work requires steps:

Promotion Championing the idea of organizational learning within an organization - without someone pushing for organizational learning it won't happen

Assimilation Collecting the facts, information, data and events into some storage mechanism often called an organizational memory [10]

Interpretation A contextual analysis and structuring of the collected data

Evaluation Assessment of circumstances and assimilated knowledge to arrive at adoptable strategies and adjusted values

Dissemination Distribution of findings Application Putting the discoveries into action

These six steps are a cycle in an ongoing process and from them we can see the process of knowledge management is a fundamental part of organizational learning.

Organizational learning is becoming a hot topic as its importance is realized. We expect a growing number of bodies and institutes like The Society for Organizational Learning [8] to be set up in the coming years.

2.4 What is the problem being solved?

As we have stressed, understanding and then accurately capturing a business process in computerized model is a very difficult task. Business processes in the main are dynamic, uncertain and are subject to unforeseen exceptions.

Workflows must be able to accommodate external factors such as changes in governmental regulation as well as internal factors such as adoption of new technology. In order to effectively support business processes in such contexts, architectures for workflow support must be both adaptable and evolutionary.

In response to these needs there is a significant and growing research effort to build more adaptive workflow systems. Special workshops such as "Towards Adaptive Workflow Systems" at CSCW '98 [2] and a growing number of relevant contributions in the business literature are evidence of this.

A valid question that so far remains unanswered is just how beneficial is it to dictate workflows. One approach to making WfMSs resilient in the face of exceptions and change is to remove the workflow control altogether - if there are no rules they cannot be broken. Such a system could still realize the majority of the benefits WfMSs offer such as quality of service and audit trails whilst leaving the control of the flow of work to the users. For example: an employee would fill out an electronic travel expense form and send it to his manager. The manager would electronically sign it and, in turn, send it to the personnel department and so on. Each participant only needs to know the next logical step in the process. Exceptions can then be dealt with pragmatically if, and when, they occur. The drawback of such a system is, of course, that the controls could not be applied where needed.

The goals of the OpenWater project are then:

- to determine a method of 'discovering' the business process by watching the process as it is being performed over a number of iterations

- to establish a method of defining workflows with minimal restraints to allow process evolution and exception handling without recourse to formal redefinition of a workflow model

- to demonstrate that a business process can benefit from the systematic feedback of historical process information to the participants of the process; that is to provide an organizational learning tool

3 The OpenWater system

In this section we first describe here the implementation of OpenWater and then move on to show how it is used to support the "Contract Sign-Off" procedure in the IBM Research Laboratory in Zürich, Switzerland.

3.1 OpenWater - a user's perspective

OpenWater can be best described as a workflow support tool rather than a workflow management system. We use the word "support" to underline that the flow of work is not dictated but rather left at the discretion of the business process participants, that is the people doing the work, wherever possible.

An electronic 'token' called an Electronic Circulation Folder (ECF) is used to represent a process instance. The ECF can be sent, in an email like way, between participants of a workflow. Unlike its paper counterpart, the ECF can be easily duplicated and sent to two or more participants at the same time. The ECF contains process data: a list of data items pertinent to the process. The data may be basic types like integer, structures, or references to complex data types like word processor documents and spreadsheets.

Users access the system using a 'browser': a simple to use graphical user interface, for both day-to-day use and administration of the system.

The browser main window, shown in Fig. 2 above has three parts: the in-tray (left), the working tray (center) and the out-tray (right). The in-tray contains a list of ECFs that are waiting to be processed by the user, the out tray contains a list of processed ECFs and working tray contains 'open' ECFs that the user is currently working on. Figure 2 shows two open

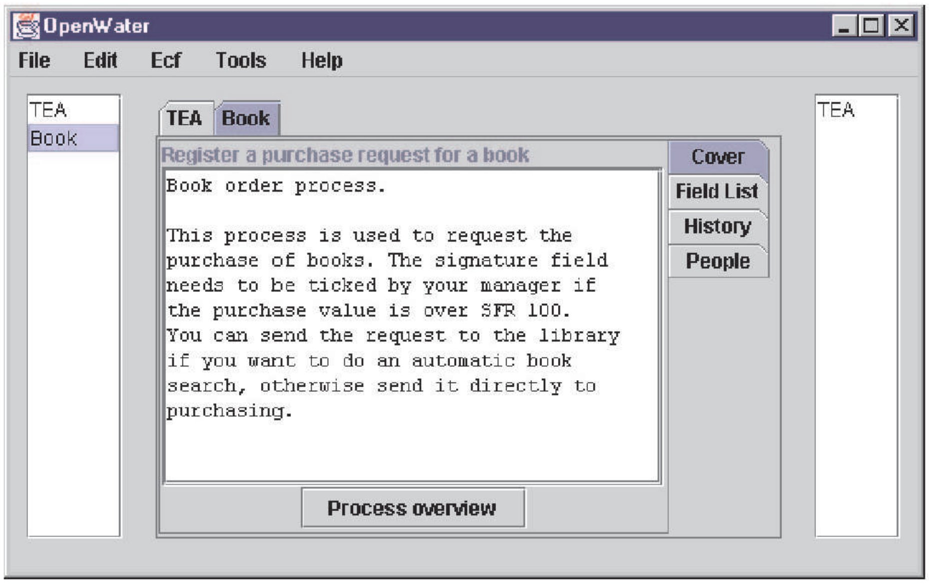

Figure 2: The ECF browser main window

ECFs: a Travel Expense Request and a book purchase request. Clicking on the tab "TEA" or "Book" brings that ECF to the top of the pile.

ECFs are represent by several pages of information. The cover contains a one line process title and optionally an HTML page containing a description of the process. The second page contains the field list - a list of the process data items like 'Client name', 'Part number' etc., these can be edited by the user. A third and fourth page contains a graphical view of the process instance history and a list of the people involved in the process.

Extensive use of drag-and-drop is used. ECFs can be dragged from tray to tray to open, close, re-queue ECFs as necessary. When ECFs are dragged into the out-tray a dialog window appears allowing the user to 'send' the ECF on to other users. OpenWater makes suggestions as to who the ECF should be sent on to - the suggestions are based on previous instances of *similar* ECFs.

The life cycle of an ECF

To illustrate how the system functions we use the example of a simple absence request process. An absence request form is filled in by the user, sent to the user's manager for signature and then to the secretary and the personnel department to update the absence list and adjust the holiday entitlement for the employee. The absence request form contains several fields including Employee name, Employee serial number, Absence reason, From date and To date.

Note that ECFs are created using an ECF template, we shall defer explaining how the ECF templates comes into being for a moment.

The user requesting the absence is also the starter of the process. The browser contains a list of available ECF templates, the user selects 'Absence'. A new Absence request ECF appears in the working tray ready for the user to complete. The user fills in the data describing the request and drags the ECF to the out-tray. The "Send ECF" dialog pops-up allowing the user to specify who should receive the ECF next. As the process is a common process within the company the system will be able to deduce the successors in the process with a high degree of accuracy - in this case the user's manager. The user clicks on the OK button and the ECF is removed from the working tray and inserted in the out tray list.

Almost immediately the ECF appears in the manager's in tray. The manager drags the ECF into the working tray - checks the dates, clicks on the manager's signature button and drags it to his out tray. Again the system guesses correctly that the next in the process is the secretary and personnel department. The manager clicks on send. And so on...

At any time the user can check the progress of the request by opening the ECF which remains in the out tray as long as the user wants. Should, for example, the manager be on sick leave, either the user or OpenWater may 'realize' that the ECF is running slower than usual and, in the system's case, raise an alarm to the user, or, in the user's case, retract the ECF and find another solution to the problem.

How do ECF templates get created?

ECF templates are first created when some person related in some way to a business process recognizes that there may be benefits in automating that process. This may be either a person who is responsible in some way for a process - a 'process owner', or often just the 'process starter'. In the Absence request example the process owner would probably be someone in the personnel department and the process starter would be the person requesting the absence.

A 'best guess' is made at the set of fields that should appear in the process data. It is unlikely that the field list will be correct the first time around, even if it is it will probably need changing later on. Template definition is almost always an on-going process.

An ECF can then be created. To see how the ECF template evolves let's imagine that, some time later, the manager decides that it would be useful to have contact telephone number for the absentee. Either the ECF, or the ECF template can be modified to include the new field. If the ECF is modified then the change is applicable to the current ECF only whereas if the ECF template is modified then the change applies to the current and all new ECFs of that type.

The manager, after adding the new field and a 'chat-note' explaining what he's done, sends the ECF back to the requestor. The requestor then fills in a suitable telephone number and sends the ECF back to the manager for signature and the process continues as normal.

How can business rules be applied?

In some cases, for instance for safety or legislation reasons, it may be necessary to enforce some business rules. In the absence request example ensuring the managers signature is just such a candidate. To support this, OpenWater provides logical rules called policies that can be applied to fields. There are two types of policy: pre-conditions and constraints.

Pre-conditions control how, by whom, and under what conditions the field is accessible. Figure 3 shows the policies for the managers signature field, it is readable by everyone and writeable by the manager of the person named in the employee field only when the fields 'From' and 'To' are not null, i.e. the have some date value. There are no constraints necessary for the signature field as it is a boolean - it is either signed or not. One might place a constraint on the To-date such as that it must be greater than or equal to the From-date.

In summary, polices are applied only when necessary. The require a level of expertise in the system over and above that of the normal user. To ease support issues the policies can be made re-usable by attaching them to field definitions in a database.

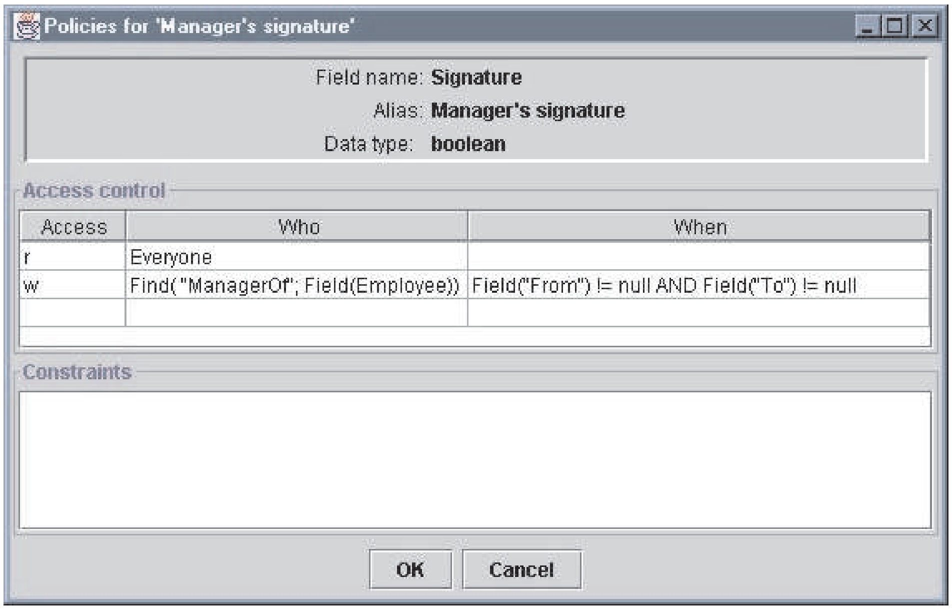

Figure 3: The policies dialog

3.1.1 Behind the scenes

Figure 4 below shows a schematic diagram of the supporting architecture for OpenWater. At the top of the diagram we can see a single process being carried out by five users of the system. The process starts at A and terminates at E. ECFs are created from the ECF factory and passed between the participants of the process. As this happens events such as data changes and ECF movements are reported to the organizational memory database (OMDB).

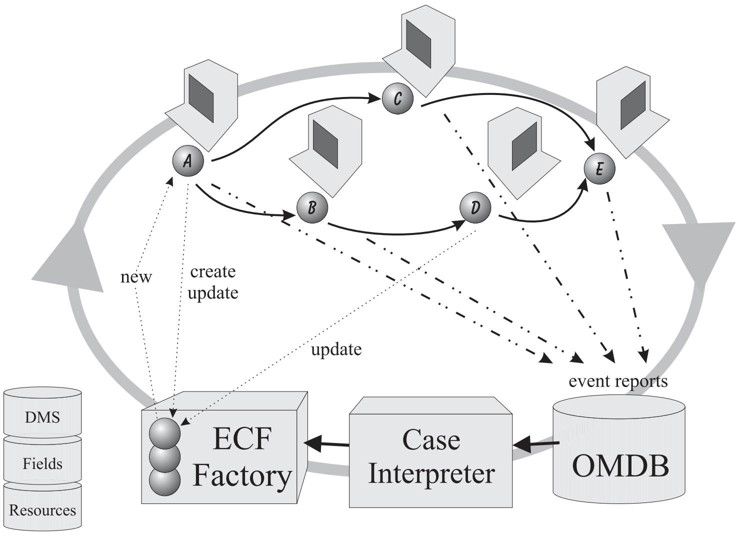

Figure 4: Behind the scenes

The OMDB stores the set of raw events. These are reconstructed by the case interpreter back into cases - an image of an individual process instance. These cases can then be used to support the user in the day-to-day performance of his work. Using this resource the user is able to quickly and easily find answers to important questions, for example:

Q : How long will this process take?

A : On average this process takes four working days with a standard deviation of one day: that is "Four days plus or minus one day."

Q : What is the likelihood that the process I am about to start will reach a successful conclusion?

A : Of 67 previous cases of this process 52 of them contain a delivery event: that is, based on previous experience, "About 78%"

Q : Did I order a disk for a PC in the demo room about 9 months ago?

A : Yes, actually two: one for "campello" and one for "pine". They were delivered on the 28th March.

It is easy to see what a powerful tool this is.

In addition to the organizational memory, several auxiliary databases are needed:

- Resources: This is the database equivalent of an organizational chart. It contains the names of people, a list of the departments and groups in the organizational and the structure and relationships between the people and groups.
- Fields: This database contains a list of re-usable fields, structures and policies for building ECF templates.
- Document Management System (DMS): this is a storage place accessible from anywhere in the organization where complex data types like word processor documents and spread sheets are stored. Each document is addressable, like a web page, by a unique URL.

3.2 The pilot - Contract sign-off

The IBM Zürich Research Laboratory, like almost every other office, is constantly undergoing a series of changes with the goal of making office life more efficient. The Contract Sign-Off process so far hasn't been automated as it was considered too complicated because the workflow changed for almost every contract. It looked difficult to beat 'good old' internal mail and telephones as technologies to solve the problem.

The current, manual, system is efficient and flexible and the clients of the process all consider it to be an adequate system. The main area for improvement is tracking and visibility; sponsoring managers (people wanting a contract to be signed) often call the contract administrator asking how the contract sign-off is proceeding and sometimes requesting that the process be expedited. The contract leaves the contract administrator with a routing slip attached and may take a couple of weeks to return - during this time it's whereabouts is not known so it is difficult for the contract administrator to respond to such questions.

At the time of writing, work is in progress to make OpenWater resilient enough to be used in a commercial environment. Once this work is complete the pilot will begin in earnest.

With the pilot we are hoping to reach several goals:

- To prove OpenWater technology in a commercial environment

- To better understand just how much control should be applied to a workflow
- To understand how to better support adaptable and evolutionary workflow capable of handling complex workflows and exceptions
- To understand how to provide historic process feedback to users and process owners and how this feedback can be used to bolster organizational learning
- To gain a deeper insight into process knowledge

4 Discussion

Platform

OpenWater is built in Java using JDK 1.2 and Swing GUI components. Data is stored via a library called ObjectSQL (shortly to be proposed for distribution on alphaWorks) in a large IBM DB2 database. This guarantees the integrity of the underlying data even in the face of a client application crash promising a very reliable system.

A question often asked is why not build on an existing platform like Lotus Notes, IBM MQ Series Workflow or Microsoft Exchange? Whilst a solution is possible using these platforms it would have been difficult and time consuming to achieve for the following reasons. Notes supports programmability through embedding a BASIC like scripting language called LotusScript in document templates. Each workflow must be programmed by someone with programming knowledge and therefore pre-researched - this breaks the OpenWater model of do-it and discover the process. In addition Notes databases are not relational - they cannot support complex data structures effectively. MQ Series Workflow again requires pre-programming a workflow and cannot support the flexibility required. Exchange suffers from many of the drawbacks of Notes and so is also unsuitable. None of these packages allow feedback of previously executed workflows to be presented to the user in a graphical form. However the results of the OpenWater project may be used in future versions of MQ Series Workflow and Lotus Notes.

Data model and security

A central copy of the process data, that is the data set connected with a single process instance, is stored in a central place and a view of this data is given to the users. It is possible for two users to modify the same data at the same time resulting in a duplicate update, rather like a Notes replication conflict. ObjectSQL handles this by versioning objects - one of the users updates will arrive first - this update is made. The second users update causes an exception to be raised as the version number change is detected. The user must resolve the conflict caused by his updates (he cannot affect other users data). Modifications to the same data item happen very rarely.

With regards to security, data visibility is controlled by logon and password. Signature fields do not digitally sign data in the current version as the participants of the contract sign-off process are considered 'trustworthy'. Later versions will incorporate such a feature. Data is stored centrally in DB2 so access can be controlled using the normal DBMS tools.

Scalability

From the point of view of the organizational memory, the more people that use OpenWater for the more processes the better. However, even in a small office, the number of on-going processes at any one time is very large. In this light it is worth bearing in mind performance issues like system response time and storage needs.

On the client side the performance needs are modest and can adequately be catered for by today's desktop computers. On the server side, storage is constantly becoming cheaper and conventional RDBMs have a good pedigree with regards to handling large volumes of data. Further, a great deal of consideration has been given to architectural and design decisions in order to guarantee adequate reliability and performance; the selection of a relational database for object persistence and lazy object loading in ObjectSQL library are examples of this. Nevertheless a sharp eye will be kept on performance issues in order to avoid any surprises.

Research or product?

As we have seen in this paper, the research content addressed by OpenWater is clear. In addition to this, OpenWater may well prove to be a valuable asset as a standalone product - the ability to support a very complex workflow like the Contract Sign-Off Process is evidence of this.

Are you telling me to sell my WfMS?

If your organization is already running a WfMS you are already aware of the advantages and disadvantages of fully automating work flows. It is likely, however, that the processes that have been implemented on the WfMS are characterized by stable and well defined workflows. It may well be that such processes are more suitable to implementation on a WfMS. The other, probably far more numerous, processes are likely to be more suitable for workflow support systems like OpenWater.

5 Conclusion

We have introduced the subjects of knowledge management, organizational learning and workflow. During this discussion a number of challenges have been identified: the need for workflow systems to adequately support ad hoc workflow and exceptions and ease the capturing of business processes in computerized models as well as the more general problems of capturing, structuring and exploiting organizational knowledge.

By defining organizational knowledge as 'experience gained in business processes' we have identified a close synergy between workflow, knowledge management and organizational learning. It seems that although this relationship has not yet been fully exploited within the business world, it looks like a very rewarding avenue to follow both in terms of research and commercial potential.

We have then described the OpenWater prototype - a Workflow Support System which supports participants of a workflow in an unobtrusive way. We have shown how OpenWater can be used to harvest data which can be used as knowledge to support future instances of workflows. And finally we have described the Contract Sign-Off Process pilot which is being used to further research in this area and to prove OpenWater as a commercially valuable tool.

5.1 The future

We are very excited by OpenWater; both from the research and technology possibilities it promises. There are several paths which looking very interesting to pursue, we present here the most promising.

Several tools and technologies are likely to emerge from the OpenWater project. IBM and Lotus could build this into future releases of existing products like Lotus Notes and MQ Series Workflow.

On the technical side there is work to do on two aspects: scalability and security. Firstly scalability: if OpenWater is accepted in a company its use is likely to grow very quickly. As a workflow system it must work reliably, a failing workflow system can have a serious impact on an organization.. As we have mentioned, a significant amount of thinking has gone into the architecture - this now needs to be backed by testing. Secondly, security is more an issue nowadays than it has ever been. It is clear that a security architecture must be put included in OpenWater - fortunately we have a great deal of security expertise in Zürich to address this problem.

In addition to the workflow and knowledge management research, several spin-off libraries and potential products have been produced such as the ObjectSQL and Expressions libraries. These are in the progress of finding their way into IBM's alpaWorks site [10].

From the research point of view we are currently looking to build links and create joint research efforts within the adaptive workflow, knowledge management and organizational learning communities. OpenWater, like a lot of new technologies, solves some problems and raises others!

And looking further afield, we'd welcome the idea of cooperating with scientists in the domains of social sciences and perhaps distributed cognition. There are a lot of opportunities to target workflow system designs at the users with the goal of improving their acceptability as tools.

References

[1] Formally known as 'Flowmark' IBM's MQ Series Workflow is a classical WfMS providing a platform to "align and integrate an organization's resources and capabilities with its business strategies, accelerating process flow, cutting costs, eliminating errors and improving workgroup productivity". Further information can be found on: http://www.software.ibm.com/ts/mqseries/workflow/

[2] The CSCW '98 conference recently hosted a workshop for adaptive workflow. The papers presenting including "An alternative approach to business process support" by Whittingham et al outlining OpenWater can be found at http://ccs.mit.edu/klein/cscw98/

[3] The Workflow Management Coalition (WfMC) is a non-profit organization promoting standards in the workflow field. They propose a reference model for supporting workflow. See http://www.aiim.org/wfmc/ for more information.

[4] Lotus Notes and Domino provide a platform for information sharing and collaborative work through the use of shared documents. Information can be found on the Lotus Corporation, Lotus Notes and Domino www.lotus.com

[5] "Working Knowledge: How Organizations Manage What They Know" provides and introduction to organizational knowledge. Davenport, Thomas H. and Laurence Prusak. Boston: Harvard Business School Press, 1998

[6] The Institute for Knowledge Management is a membership alliance made up of senior executives from major corporations, together with knowledge management experts from IBM, Lotus, industry, and the academic world. The Institute conducts research and provides leadership in the field of knowledge management. http://ibmpnyx1.palisades.ihost.com/ikm/ikmhome.html

[7] The WWW Virtual Library on Knowledge Management contains a large number of links to references on knowledge management. It can be found at http://www.brint.com/km/

[8] The Society for Organizational Learning is a global learning community dedicated to building knowledge about fundamental institutional change through integrating research, capacity building and practice. More information at: http://learning.mit.edu/

[9] "Organizational Learning" is both the starting point and reference for the subject. Argyris, C., Schön, D., Organizational Learning: A Theory of Action Perspective, Addison-Wesley, Reading 1978.

[10] There are several works on organizational memory particularly by M. Ackerman one of those is Augmenting the organizational memory: A field study of Answer Garden. In: The Conference on Computer Supported Collaborative Work (CSCW'94), New York: ACM. Pp 243-252. **Miscellaneous** IBM's alphaWorks site is an online laboratory for next-generation Internet technology directly linking the technologist and Internet research communities. For more information visit www.alphaWorks.ibm.com