The Story of a Working Workflow Management System

Steen Brahe
Danske Bank and IT University of Copenhagen
Holmens Kanal 2-12
1092 Copenhagen K, Denmark
stbr@danskebank.dk

Kjeld Schmidt
IT University of Copenhagen
Rued Langgaardsvej 7
2300 Copenhagen S, Denmark
schmidt@itu.dk

ABSTRACT

This work presents experiences with the adopting of a workflow management system in a large financial institution. We describe the gradual evolution of a traditional work process, from manual to computational regulation of coordination. The study shows that computational workflows may increase labor productivity remarkably and in general have significant economic benefits, but also that to make current workflow technologies yield such results requires unorthodox twists and tweaks in the workflow design to allow for worker control and process overview. The paper argues that workflow technology is now at a stage where the contribution of CSCW is obviously needed.

Categories and Subject Descriptors

D.3.2 [**Programming languages**]: Language Classifications—*Very high-level languages*; H.4.1 [**Information Systems Application**]: Office Automation—*Workflow management*; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Computer-supported cooperative work, Organizational design*; K.4.3 [**Computers and Society**]: Organizational Impacts—*Computer-supported cooperative work, Reengineering, Automation*

General Terms

Design; Human Factors

Keywords

Service Oriented Architecture (SOA), Business Process Management (BPM), Workflow Management (WFM), BPEL, case study

1. INTRODUCTION

The issue of what role, if any, computational systems might play in the regulation of the flow of tasks and documents among cooperating actors and organizational units has been a constant theme in CSCW from the very start.

In fact, CSCW can arguably be seen as emerging, at least in part, in reaction to the disillusionment that arose within the Office Automation community in the mid-1980s. Until then, hopes had been high, encouraged by the bright prospects offered by the new techniques for constructing computational models of highly distributed systems that had been developed since the early 1960s, especially on the basis of the graphical modeling notation developed by Petri. With the emergence of networked computers with graphical user interfaces, it seemed obvious to explore whether and to which extent such representations might be exploited as a means of modeling and regulating "office procedures" and other kinds of workflows [18, 6, 7]. The high hopes were quickly frustrated. Workflows were not as easily captured and modeled as had been anticipated. "Exceptions" turned out to be an everyday occurrence in "office work", and as soon "exception handling" mechanisms had been devised, analysts had to realize that there are also exceptions to exceptions. In short, the status of "office procedures" turned out to be problematic [14].

This realization motivated multiple lines of CSCW research. For some, this experience meant that the very idea of computational regulation of workflows was impossible (or unacceptable), which in turn motivated the development of alternative forms of computational support of interaction (e.g., [10, 11]). The research on "media spaces", "collaborative virtual environments", "shared work spaces", etc. can be seen as extensions of this line of research. Other researchers instead explored ways of making workflows adaptable (e.g. [8]), e.g., by providing ordinary workers with environments for building and modifying computational workflow models for their own use [13, 4]. Parallel to this, of course, a line of research has focused on investigating the ways in which workflows are actually devised, constructed, adapted, modified etc., in cooperative settings (e.g. [3]). All this is now part, albeit a small part, of CSCW history.

However, while all this has been going on as a sub-theme in CSCW, workflow system technology has been a topic of intense interest outside of CSCW.

Since the 1990s, Workflow Management (WFM)[1] has received considerable attention from industry as well as from a research community interested in the technical aspects of workflow management [15, 17]. In the late 1990s, several commercial WFM systems became available but were not widely adopted [12]. This can be explained, to some extent, by lack of infrastructure technology to integrate legacy sys-

[1]The term Business Process Management (BPM) is now often used instead of Workflow Management [16]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GROUP'2007 2007, Florida USA
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

tems and new applications effectively. Recently, a new enterprise architecture, named Service Oriented Architecture (SOA), has emerged as a promising foundation for WFM systems and is now becoming widely accepted by the industry. In fact, most of the contemporary WFM systems build on a Service Oriented Architecture. SOA is an enterprise architecture that advocates loosely coupled and reusable systems and has evolved from component-based development and distributed Internet architectures as a new abstraction layer that allows internal and external systems to interact using common standards and protocols. With SOA, systems developed on different platforms and technologies, such as, e.g. legacy systems, Java, and .Net applications, are able to communicate directly using standardized interfaces and protocols. In short, SOA makes it possible for an enterprise to open up its legacy systems to other systems and services.

A WFM system — sometimes also called a "workflow engine" — supports the definition and execution of "business processes" or "workflows" in a common workflow language such as the Business Process Execution Language (BPEL)[1] that was defined in a joint effort involving large companies such as IBM, Microsoft, and Oracle. BPEL is a high-level language that is used to construct computational or executable workflow models and is based on "orchestration" of "services", i.e., (remote) computer systems that can be invoked to perform some task, into complex applications. Hence, BPEL and related languages are often mentioned as "service orchestration" or "service composition" languages. (In the remainder of the paper the term "workflow" is used for a computational workflow model implemented by means of a WFM system).

Several of the current WFM systems that are based on BPEL extend BPEL to allow for the definition of workflows that combine execution of automatic services and manual tasks handled by humans. Such extension involves two parts of the WFM system. First, the language is extended with a new task type, e.g. a HumanActivity, or a Staff type, which allows the modeling of workflows containing tasks to be handled by humans. Second, the WFM system must contain some kind of task portal, where process participants can "claim" and "execute" tasks assigned to them.

A new outline standard BPEL4People has been suggested to address this issue [9] but still has a long way to travel. BPEL is currently the most widely accepted industrial standard and is a hybrid between IBM's graph-based WSFL and Microsoft's block based XLANG which have their origin in Petri nets and phi calculus respectively.

In short, WFM systems constitute an architectural layer on top of SOA that affords the composition of services and people into complex workflows with automated coordination of tasks. These workflows are able to "execute" and coordinate business processes. This is illustrated in Figure 1.

What may be of particular interest to the CSCW community is that WFM systems and workflow technologies such as BPEL have now evolved to a stage where they are been deployed and used in large-scale work settings. That is, instead of discussing computer-support for the regulation of interdependent activities as a interesting but hypothetical proposition, we can now begin to investigate how they are developed, deployed, appropriated, amended, used, redeveloped, etc., "in the wild".

In this paper we will tell the story of the stepwise development of a business process using a WFM system that is

Figure 1: Business processes as a composition of services and human tasks.

not only actually working but also has turned out to be surprisingly successful. We will sketch the development process and will try to bring out some of the sources of the successful implementation. In doing so, we will focus on the technical issues workers have experienced in the course of the development of the system and especially on the issue, raised by workers, of control over the execution of workflows.

2. FROM MANUAL TO AUTOMATIC COORDINATION

Our story is the story of two separate and large projects both concerned with implementing cross departmental processes in a large Danish financial group, Danske Bank, which we have investigated.

Founded in Copenhagen in 1871 as "The Danish Farmers' Bank", the group has grown to become the largest financial group in Denmark and one of the largest in northern Europe. It now provides a variety of financial services such as banking, mortgage credit, insurance, pension, capital management, leasing, and real estate agency.

Danske Bank has implemented a WFM system based on a commercial product from IBM, which has been extended in areas where the Bank's business requirements were not met. The bank has for instance created its own task portal to achieve higher flexibility in allowing allocation of tasks to different process participants and to achieve better overview of running workflows. Business processes are implemented as workflows using BPEL. The Bank's WFM system depends on SOA, which it adopted back in 2001. All new application development is targeted to SOA and legacy systems are "service enabled". This allows all applications to be executed from within workflows. In other words, applications and services developed for one part of the group can be located and used by other parts of the group through a central service library and repository. Currently, the group has several thousand different service operations.

2.1 Empirical Investigation

In the last four years the first author has been a member of the team responsible for implementing Danske Bank's WFM system and has had daily contact with the developers responsible for the primary project described in this paper

("Customer Package"). Eventually, a systematic investigation of the project was conducted over a period of about five months.

The systematic investigation involved interviews with a project manager, with developers of the workflow system, and with solution architects and a back-office worker. We also held a half-day workshop with the WFM department and the developers involved in developing the workflow systems, at which we discussed different issues related to the WFM system. All interviews and the workshop conversations were recorded, transcribed, and analyzed.

In addition, the documentation of business process models and the workflow implementation code were investigated.

During the five months of systematic investigation, we had "coffee"/corridor discussions on a weekly basis with people engaged in the examined projects.

2.2 "Prehistory"

Our story begins in 2003. But the prehistory needs to be told first. Until 2003, when a customer entered a branch of Danske Bank to open a new account or purchase an additional "product" or service, such as a credit card or Internet access, the customer had to sign a number of documents. For each product the customer requested, one document had to printed, read, checked, and signed. As customers on average would request seven products, a large number of pages had to be printed and signed. In fact, a customer could leave the branch with up to 100 pages! This process was obviously not satisfactory to the customer, since he or she would have to spend quite some time in the branch, often more than an hour. The large pile of paper was of course also confusing. To the bank clerk, the "customer advisor", it was not satisfactory either. Printing the many pages was time consuming, and after the customer had left the branch, all "products", i.e., contract specifications, had to be created manually by means of different programs (figure 2). All in all, creating products for an average customer could take up to several hours.

Figure 2: Customer advisors create all products manually.

At some point in 2003 the Credit Card Department wanted to increase the sales of American Express credit cards. However, the then existing procedures, according to which several documents, such as agreement about the card and exchange of information, would have to be printed and signed, were seen as an impediment to this plan. But one day somebody at the Business and IT Development Department proposed to devise a generic contract where different products were bundled in "packages". Different packages were defined to target different customer segments and each contained default and optional products. The concept of "Customer Package" was born (or rather reborn, since the notion of packages had been used earlier in one of the branches with success). When a customer now visited a branch, he or she would purchase a "customer package" containing, say, an account, a credit card, and Internet access, and in doing so, only one document covering the whole package would be printed, filled in, and signed. The new sales concept was introduced in June 2003.

Having analyzed the sales and product creation process in the branches, the Business and IT Development Department initially decided on a solution without any IT support. The reason was to be able to introduce Customer Packages to customers as quickly as possible, instead of having to postpone market introduction until IT support had been developed and made ready for use. Instead, the solution that was adopted consisted in a standardized Microsoft Word document, a template, which could be customized by the individual branches for greater flexibility. Using the template, the adviser could specify all products for a particular customer in just one document. Compared to previous practices, the customer had only one document to sign. Furthermore, the time spent on the sales process was reduced. Although extremely simple, the template was a success in that the number of credit cards as well as other products that were now sold increased significantly. And it was easier to introduce different products.

The successful adoption of the Customer Package template then triggered a series of organizational transformations that fit beautifully with Babbage' classical schema: from traditional craft work, to systematic division of labor, to stepwise automation of individual processes, to more or less automatic control of the overall process as a whole.

2.3 From Division of Labor to Automatic Validation

While customer experience improved markedly with the Customer Package template, the customer advisors still had to use a large part of their time on creating products after the customers had left the branch. Because of the time-consuming task of creating the products, it was difficult to find time to meet with customers. Someone then asked, "Why can't we use our time on advising customers and selling products. Someone else can create the products". The Business and IT Development Department agreed, and in August 2003 a new back office department named "Customer Agreements" was established to handle the creation of products that were ordered by means of customer packages. Under the new arrangement the customer advisor, after having obtained a signature to the customer package document, would send the document as an email attachment to the Customer Agreement Department. Here, a staff of about thirty people worked on these documents: when they had received them, they would read them carefully and then manually create the products by using different systems (figure 3).

For the customer advisors the new back office function was a satisfactory arrangement, as they could now use their time more effectively on advising customers and selling products. However, in some ways the problems recurred in another shape and at another place. To the back office workers the documents they received from the customer advisors posed multiple challenges; the data were often misspelled or contradictory or the advisors had forgotten to provide required information. The back office workers thus had to carefully read through the entire customer package document in order

Figure 3: Backoffice workers create all products manually after receiving word document from customer advisors.

to check and validate it, and they often had to phone the customer advisor who had created the document to get the correct information so as to be able to create the products. Furthermore, in addition to being time consuming and tedious work, the procedure of validating documents and communicating with the advisors had not been reconsidered and reconceived. Each worker had separate ways of validating the document.

Unsurprisingly, a strong need was felt for improving the process of handing over customer packages from customer advisors to the Customer Agreements back office. Therefore a new system, called CTS (Case Transfer System) was created in Lotus Notes. This system allowed the customer advisor to choose customer type and thereby obtain a new document for this specific customer type, and to then send it to the Customer Agreements Department immediately after entering data about the customer and the ordered products. The system would extract the data from the Word document and make an automatic validation check on the entered data such as presence of required information, marked checkboxes, etc. In case of data conflicts, the CTS system would return the document to the customer advisor without involving the Customer Agreement Department (Figure 4). This was a clear improvement over previous practice. For

Figure 4: Backoffice workers create all products manually after getting validated word document from the Case Transfer System.

the advisors, it was an improvement in as much as they were notified directly about errors in the document and they were able to correct the errors at once, and it was an improvement for back office in as much as they received documents with fewer errors. Furthermore, the document was now trimmed of unnecessary data, so that it was easier to find the relevant data. With the new solution, it was no longer necessary to read the entire document in order to ensure validity.

2.4 Process Automation by Workflow Management

The process was now "streamlined" not only for customers

Figure 5: Workflow enabled backoffice process with backoffice workers creating seperate products.

and customer advisors, but also to some extent for the back office staff. However, the product creation procedure had not been adapted, merely transferred. The back office basically handled the product creations in the same way as the customer advisors had done previously. The different products of a package were created in a different sequence from package to package, although there might have been a more rational sequence in which to create the products. Furthermore, the same data about the customer, accounts, etc. had to be entered manually in the different systems that had to be used in order to create products. This often resulted in misspelled data and was in any event still quite tedious work, since the same data had to be entered again and again.

As mentioned earlier, at the end of 2001, the Bank had decided on and implemented a Service Oriented Architecture to which all new IT development would be oriented. In addition, by the end of 2003, the company was ready to implement business processes as workflows based on "service orchestration" and BPEL.

At that time, it so happened that the people from the Workflow Department who would be in charge of building computational workflow models, or workflows, by means of the WFM system were placed physically next to the people that developed the CTS system. In discussions between these two groups of people it was soon realized that the process of creating products for customer packages was ideally suited for exploring and demonstrating workflow technology. It was a highly predictable and production-like process, and it involved systems from several departments throughout the group. A workflow for this cross-departmental process would be able to link all the different systems together and eliminate the need for entering the same data repeatedly. It was therefore decided to use the manual customer packages creation process as a pilot study for the new WFM system.

As the first step, the CTS system was extended so that it could extract data in XML format from submitted Word documents and to initiate instances of the workflow using the extracted data. Figure 5 illustrates the updated scenario. An extended validation function was also implemented in the CTS system. After having made the XML extract, the validation function would determine if the package was of a character that was unsuited for handling by the workflow. In such extraordinary cases, instead of using the workflow, the particular package would be assigned for manual treatment.

The workflow was put into production in December 2003. The first version of the workflow was implemented precisely

in the way the back office workers used to create products. All products (except one) were still created manually by means of the same systems as before. But now, in December 2003, the workflow automated the allocation of the different tasks and the determination of the sequence in which they should be carried out. Based on the document filled-in by the customer adviser, the workflow would create a list of tasks — or products to be created — for the back office workers and display this in the WFM systems task portal. A back office worker logging onto the task portal would now simply see a list of pending tasks. These tasks would be organized according to priority and relevance for the worker. The sequence in which the tasks should be executed would not be prescribed and in addition, the tasks would be from many different customer packages. Only few tasks for a specific customer package would be delivered immediately to the task portal. The remaining tasks would appear after execution of the first tasks. For instance, a task for ordering a credit card pin kode would first appear in the task portal after the credit card had been created. When accepting a task, the worker was automatically transferred to the application program relevant for creating the given product, and available data were made available, delivered by the workflow. The workflow thereby became the glue that bound the different systems together.

The WFM system constituted another improvement of the product creation process, because the system made the relevant data available to the back office workers and guided them directly to the relevant systems from the task list.

Now, having implemented the business process as a computational workflow, the Business and IT Development Department began to look for "optimization" possibilities. The obvious optimization of a workflow of this kind, largely a "composition" of existing computer programs, is to automate the manual tasks of launching and transferring data between the different programs. The Business and IT Development Department started to contact the departments responsible for the product systems and requested automatic product creation services. Unfortunately, most of the departments did not have the resources required for developing the services in question. However, the Accounts Department agreed to develop an automatic account service to handle an important product that always had to be created for a customer package, namely, an account. The automatic service was incorporated into a new version of the customer package workflow. The back office workers now did not have to create accounts; this was handled automatically by the workflow and the new automatic account service. This was an eye opener for the Business and IT Development Department; "If we are able to automate the creation of accounts, then we will also, through systematic work, be able to automate much more of the product creations".

Another obvious optimization of the workflow is to implement what are called "exception cases", that is, cases that are not exactly "exceptions" but rather extraordinary cases. In the course of the ongoing "business process optimization", previously manual activities have been optimized so that they can be performed by automatic services. However, in the first version of such an automatic service, for instance the "creation of cards", i.e. the issuing of credit cards, it has been the experience that it sometimes fails and has to be taken care of manually. As for the credit card process, this has turned out to be the case in 4% of the instances. Instead of a process that is interrupted on a regular basis due to such annoying failures, such instances are now weeded out automatically, by the workflow, based on an analysis of the data. Thus, according to the data in the given instance, the workflow assigns the task either to the automatic card creation service or to the back office workers. This approach has also been adopted by the customer package workflow. One of the first tasks in the workflow is a data evaluation task. Here, the data from the customer is evaluated, and in complex cases which have not been implemented in the workflow, the process is sent to manual execution without workflow control. The partial implementation approach, where 80-98% of all case are handled, has been a successful strategy. It is cheaper and faster to implement a system that only takes care of the majority of the cases and leaves "exception cases" to manual handling. However, over time one "exception case" after the other have been "optimized. Thus, while many exception cases initially were caught and assigned to manual execution, over the years, more and more "exception cases" have been implemented and can now be handled by the workflow.

2.5 Extending the Business Domain

In the autumn of 2006, the concept of Customer Package and the concomitant business process and workflow was extended to include the creation of "single products", that is, supplementary products. When a customer advisor orders an additional product for a customer, for instance an additional credit card to supplement his or her package of products, the customer advisor previously had to create the card manually, but the order can now be submitted through the CTS systems. The CTS system instantiates a customer package workflow which either creates the ordered product automatically or assigns the order to a back office worker.

Although it should be possible to automate more than 80%, complete automation of the process is unlikely. Further steps will require system owners to develop automatic services, and this has been a bottleneck due to lack of resources. It is difficult and time-consuming work to prescribe and implement all exception cases in a workflow, and the departments have other things to do that have higher priorities. At the end of the day, it is a question of balancing costs and benefits and of what economists call "diminishing returns". Although further optimization will include implementation of additional exception handling features, so that fewer packages have to be assigned for manual execution, it may simply not be economically sound to try to automate the more exotic "exceptions". The challenge of handling the challenge posed by such recalcitrant exotic exceptions has been addressed by letting the CTS system analyze the XML data extract from the Word document. If the particular package is evaluated as not suitable for the workflow, the Word document is assigned to the back office for manual execution.

Figure 6 illustrates the percentage of all daily received customer packages that are handled by workflow. In the first versions of the workflow in early 2004, the majority of packages were assigned to manual execution, as only very few exception cases had been implemented. The workflow was then optimized, and an increasing part of the total number of packages was handled by the workflow. However, in 2005 and 2006 the customer package concept was introduced in Sweden and Ireland where the workflow was not

Figure 6: Percent of all customer packages handled by workflow.

Figure 8: Number of customer packages per day.

implemented, and as a result the rate of handled processes handled by the workflow decreased. But in late 2006, three initiatives increased the percentage: several exception cases were implemented in the workflow, initiatives were taken to force branches to use the CTS system, and the workflow was extended to be able to create single products without any customer package. The latest version of the workflow is able to handle most exception cases, and today only a small part is assigned to manual execution.

The percentage of packages handled by workflow, as illustrated in Figure 6, is calculated relative to the total number of packages, including other brands and languages which as yet cannot use the workflow technology. Had the number of packages handled by workflow been calculated relative to the number of packages for which the workflow has been designed, but which were nonetheless sent to manual execution by the CTS system, the percentage would have been higher.

2.6 Results so far

For about two years, the Customer Package workflow has systematically been improved and optimized by addressing the most expensive and time consuming tasks. In the first version of the workflow, all tasks were handled manually. Today, the workflow is running in version 6, and about 80% of all the products are created automatically (Figure 7). The back office workers have saved a significant amount of time which they now use for processing more packages as well as for other activities.

Figure 7: Workflow enabled and optimized backoffice process with 80% automated product creation.

From the time of introducing the workflow in December 2003 up till May 2006, the number of customer packages amounted to about 300 per day. Between May 2006 and May 2007, this number has grown by a factor 6 to about 1800 a day. The explosive growth has two reasons. First, until the summer of 2006, several branches did not yet use the CTS system to submit customer packages. That is, they created products locally as before, without using the back office. During the summer of 2006, initiatives were taken to simply force branches to submit customer packages through the CTS system. Second, the customer package workflow was extended to allow creation of "single" or supplementary products.

Figure 8 illustrates the total number of customer packages per day; it includes packages handled manually for all languages and brands and packages handled by the WFM system. The fluctuations in the diagram have different causes, such as, say, summer holidays, introduction of customer packages in other brands, and marketing and new product offers for the customers.

The diagram indicates that the effect of forcing branches to use the CTS system and using the customer package workflow to create single products has been substantial. However, the diagram should be read with reservation. First, over the last four years the number of back office workers as well as their responsibilities have varied, which makes comparison over time somewhat problematic. Second, it goes without saying that the creation of single products requires much less work from back office compared to a "real" customer package, as only one product has to be created for each "case". Therefore, the growth in number of customer packages by a factor 6 has not caused a similar growth in back office workload. In sum, Figure 8 only indicates a large growth in number of packages but nothing about labor productivity.

However, during the first period of working with the WFM system, some performance measurements of the back office workers were made. Measuring the amount of time used on creating a single product when working with a Word document or with the WFM system, it was found that the workers were about 19% faster when creating products delivered through the WFM system. This effect can primarily be attributed to the fact that a manual task delivered by the workflow contains a direct link to the system in which the particular product is to be created, and that available data is provided directly to the system.

In sum, the automated coordination of tasks and the continuous optimization and automation of product creation means that back office workers today are able to handle between two and three times as many customer packages as

(a) Main customer package workflow.

(b) Main customer package workflow printed on 24 A4 pages to allow a better overview.

Figure 9: The 6th version main product creation workflow.

when working with Word documents. In 2003, the performance requirement for a back office worker was about 20 packages a day. In 2007, the requirement is 50 packages a day.

3. TAKING CONTROL

Participants involved in defining, implementing, and executing the customer package workflow have all had issues with the concepts and technologies of a WFM system. We will here highlight some lessons learned related to the back office workers.

To back office workers the workflow has had obvious advantages, not least due to the fact that they, when accepting a task in the task portal, were directly linked to the responsible application program, by means of which they were able to handle and execute the task, and that the available data were transferred. This was considered an advantage, as they did not have to access the application program and enter data manually.

However, being the human component of the automated customer packages process was a new way of working. From being in control of the complete business process of creating all products for a customer, they now were allocated fragmented tasks to be executed from the WFM system's task portal, without any knowledge of the context in which the tasks originated.

The WFM community typically presumes a "task-centric business process" model in which workers do not know and do not need to know the business process and should only be concerned with handling discrete tasks. It is supposed to offer the advantage that individual tasks can be distributed to different people and departments and that management thus is able to control the allocation of resources. However, the task-centric way of working was not accepted by the back office workers; "We do not want to be a factory. You get better caseworkers when they follow complete cases and not separated activities". Their reason for not accepting to be assigned discrete tasks is that they would loose the overview of the case if they worked task-based. This is considered especially problematic in the handling of "exception cases" where overview is a necessary precondition for figuring out what caused the exception and how the case can be dealt with.

Consequently, the back office requested a change to the design of the workflow so as to be able to work in a case-centric matter. To meet their demand, some interesting twists and tweaks were added to the design of the workflow. An additional manual task — a dummy task, really — was appended to the workflow as the first one to be executed in each instance. When a worker would claim and execute this task, all remaining manual tasks pertaining to the same package would be allocated to the same worker. In short, the back office worker reclaimed responsibility for executing all tasks related to a particular customer package.

In addition, the workers are able to monitor, via the task portal, manual activities that have already been executed. They have also expressed a wish to be able also to monitor automatically executed activities, since this would facilitate better understanding of what has already been created in the given process instance.

This was not the only situation where the issue of control of the workflow emerged. As noted in passing above, we have also investigated another project, the Account Settlement project, which implemented a similar business process as a workflow. This workflow, which terminates a customer's engagement with the group, has interesting similarities with the customer package workflow. Although front office and back office workers participating in this workflow have no contact to back office workers of the Customer Agreement Department, they have independently of the others also found ways to work in a case-centric manner. In the Account Settlement workflow, workers focus on one customer case; through the task portal they locate all activities that belong to the particular case and handle them all before starting on the next case.

Although it is against the precepts of prevailing WFM systems philosophy, the workers have their reasons for insisting on working in the case-centric manner.

First of all, unsurprisingly, according to the back office workers, "it does not feel right" not to know the larger context in which one is working. Knowledge of the process gives one a better understanding of what one is doing, and handling all the tasks pertaining to a given case gives "a better feeling of doing a complete piece of work". In addition there are some very good practical reasons. When one worker han-

dles and has the responsibility for a complete case, with all its associated tasks, this has the advantage that the worker knows what has happened during the execution of the workflow, and the worker should be suited to handle exception cases and failures faster.

4. SOURCES OF SUCCESS

According to an old Russian proverb, success has many fathers, while failure is an orphan. This undoubtedly also applies to our story of the working workflow. The sound strategy of aiming at a realistic level of automation and leaving the remainder of cases to manual execution is definitely one factor. The tweaking and twisting of the workflow design allowing the back office workers to retain control of all tasks pertaining to each particular case is surely another reason why the development process has achieved such significant results. A number of other factors that have contributed to the success are worth mentioning.

4.1 Motivated workers

The back office is primarily staffed by bank clerks who have expressed an interest a job "in the rear", that is, away from the pressures of the front office, as well people from post offices who have been used to high-volume keyboard work. The average age of the back office staff is about 55 years. Common prejudice notwithstanding, the workers have exhibited strong motivation for using the WFM system. This can primarily be attributed to an enthusiastic group within the office, who expressly wanted to try something new and has been the driving force in adopting the system, as well as the group leader who from the very beginning said that "this is the way to go". There is today a general confidence about the WFM system and the back office workers would not want to change back to the older CTS system, where regulation of the process was handled manually. They have acquired an understanding of the WFM system and express satisfaction with the advantages it offers in terms of coordinating processes as well as automation of manual tasks. One of the workers commented that "It is strange to think that a large part of our former working day today executes automatically."

4.2 User involvement

From early on, back office workers were directly involved in the development of the customer package workflow. Both developers and workers report that they found out that they could learn much from each other and talk of the cooperation as "a good experience".

For the developer, the close collaboration made systems integration smoother, as he developed an understanding of the practical business process, of how people worked, and of how to use different business systems. The workers could simply show which application programs were used for which different manual activities, which data should be provided, and what were the preconditions. Much of this information had been lost under the traditional development model of transferring information from worker to business analyst to architect and to developer. On the other hand, the workers expressed that they were involved and had influence. In addition, they acquired an understanding of the complexity of the involved systems, which made it easier for them to accept periodically unstable systems as well as to have patience with the time it took for the workflow to develop.

Figure 10: Developer and back office worker cooperating during development and test of the customer package workflow.

The back office worker at Figure 10 declared: "The developer and the user really have learned a lot from each other. It is very very important to make the developer, the user, and the business analyst communicate with each other early in the project. It is not sufficient that the developer talks to the business analyst. The user also has to be involved... Take some concrete examples and work through them so that everybody agrees on how the business process works."

Today, the workers have adopted the new way of working, and despite periods of unstable systems, workers express agreement that they have become much more efficient than with previous work practices.

4.3 Stepwise refinement

The development and implementation process has been gradual. The customer package process has been deployed in the six major versions. In addition, several minor versions have been deployed with minor changes and error corrections. This stepwise refinement process and the high level of user involvement it affords have probably been a precondition for the workers' strong commitment to the process. As already noted, the first version of the customer package workflow was implemented exactly as the then current manual work practice. This was thought to make the transition from manual to computational workflow easier. Thus, although the back office workers had to learn a new system, the underlying business process was well known to them. Since that first version, the customer package process has been refined in stepwise manner. This has made it possible for all parties to learn from experiences and to have the lessons learned reflected in the changing workflow. On the other hand, the stepwise development and implementation process would hardly have been possible, not to mention economically viable, without the relatively high degree of flexibility afforded by the combination of a high-level "service composition" language such as BPEL and SOA.

5. ISSUES WITH CURRENT WORKFLOW TECHNOLOGY

Now, although the gains in productivity and business flexibility that have been obtained are remarkable, experience with current WFM technologies show that they can not be considered fully mature. Some of the major challenges, as seen from the back office, are described in the following sections[2].

5.1 Performance and System Stability

The first issue is not one of technological principle but still an issue of practical import in any real-world setting. During the first phases of introducing the workflow, users experienced long response times and unstable systems. Today, the long response time has been resolved, but periodically systems still become unstable. Now, the performance of the back-office workers is measured on how many packages they handle, so when systems become unstable or crash, the workers are not particularly happy, to put it that way. During the first period of using the WFM system, the back office workers were not even informed about phases of system instability, but procedures have by now been defined for informing workflow users about potential problems with system stability.

The worst incidences of this kind were related to activities that had been automated but subsequently turned out to be error-prone. For instance, four day after the card creation task had been automated, it was recognized that the card creation system did not function correctly. As a result, the back office workers had to go back and create cards manually for customer packages from the last four days as well as for new customer packages that were still arriving.

The system instability that has been experienced has mainly been caused by bugs in the WFM system or in related systems such as database and messaging systems. These bugs seem to indicate that not only is a WFM system an extremely complex system complex but also that the vendor is still developing it.

5.2 Costs of Changing Workflows

Although BPEL in combination with SOA affords a remarkable increase in workflow design flexibility, we are far from a situation where ordinary workers are able to define and compose their own local computational workflows. The technology requires the ongoing intervention of highly trained technical specialists. And the development process requires great effort and takes time.

An example will suffice to make the point. At some occasion, the developer was contacted by the business analyst who wanted him to make a change in the process; an additional manual task had to be added to the workflow. The developer thought it was a strange change request but updated the workflow according to the request and tested it in three test environments before deploying it to production. This was a time-consuming process, as the complete workflow should be tested thoroughly. Shortly after deployment, however, the back office workers started to phone in and asked about the new manual task; it was unnecessary, they said, and they wanted it removed. The developer immediately removed it, but it took almost two months before the next version was deployed. And in the meantime, the workers had to claim the unnecessary manual task for all new customer packages.

5.3 Cross-departmental Dependencies

A workflow such as the customer package workflow integrates systems from multiple departments. However, these departments are not aware that the workflow depends on the availability of their systems. A couple of times it transpired that other departments were going to change or decommission systems on which the customer package workflow depended. If that had happened, if the systems are changed or closed, the workflow would not be able to execute, and the back office would grind to a halt. An example: the workflow depended on a traditional 3270 mainframe system for creating payment authorizations that allow the bank to pay customers' bills. One day, however, it was announced over the intranet that this legacy system would be closed and replaced by a web based system. However, one of the back office workers noticed the problem and immediately contacted the responsible department: "You cannot close the system from day to day, the workflow department has to be able to catch up and change their workflows to use the new system". As a result access to the old system was prolonged until a new version of the workflow was ready. The example is of course just an anecdote and even an anecdote with a happy ending, but the point is no less valid for that: workflows on top of highly distributed architectures such as SOA creates increasingly tightly coupled dependencies across organizational boundaries and will thus, potentially, contribute to increased complexity. Mature SOA implementations will have to support the distributed cooperative management of these dependencies.

Software vendors have started to implement such dependency management systems, e.g. IBM with its "WebSphere Service Registry and Repository" [5]. However, implementing and integrating such a system into the existing infrastructure of a large-scale enterprise is a difficult, exacting, and protracted task.

6. DISCUSSION

We have shown that workflow technologies are now at a stage where they begin to be used for real in large-scale settings. We have shown that the technology, while not completely mature, affords a relatively iterative development process and thus the possibility of learning from experience and of involving users in the design process. And we have shown that the workflow technology holds potential for remarkable productivity gains. All this is quite remarkable and of obvious interest to the CSCW community.

We observed that workers claimed control over the workflow to be able to work on complete cases, in contrast to handling separate tasks for multiple different and hence fragmented cases. They needed a deeper understanding of the context they worked in. We also found that an iterative development of the workflow system, in the context of highly motivated workers intimately involved in the process of defining the system, were some of the reasons for the success of the WFM system.

However, we do not want to exaggerate the general validity of our story. First of all, in spite its scope and complexity, banking, as a work domain, basically deals with a world of

[2]Challenges experienced by the business analyst, the solution architect, and the developer when developing the workflow are described in [2]

relatively well-behaved signs. A quick comparison with work domains such as medical work, construction, manufacturing, etc. should suffice to make the points: signs can of course be wrong, mistyped, misplaced, incomplete, etc., but in contrast to ailing patients, solidifying concrete, or steel under stress, they rarely behave in a contingent manner. It is, of course, therefore no accident that accounting work has been pliable to automation.

Second, the business processes we have been investigating were selected as workflow pilots precisely because of their predicable and production-like character. They were relatively well-behaved in advance of the development effort. Moreover, the processes that were selected were already, to a very large extent, computerized from the beginning, in the sense that they depended on an array of application problems, although these application programs required human intervention to be stringed together. To a large extent, then, the purpose of the workflows consisted in the "orchestration" of already computerized processes. One would not easily find processes like that in medical work, construction, or manufacturing. Consequently, the task portal used by process participants offers only rudimentary support for cooperative work.

That is, the success of technologies like BPEL as described here may not be easily repeated in other domains. Anyway, the story does show that workflow technology is coming of age (despite technical and organizational issues with system stability and dependencies between services). As the example of BPEL4People indicates, this raises new prospects for CSCW. The point we want to make here is this: BPEL and similar languages treat human activities on par with automatically executed processes. For the workflow, a manually performed task is no different from an automatic service: they are both simply black holes. However, when human activities consist in far more than retrieving and entering data from and to application programs and are thus vastly more contingent, then the black hole assumptions break down.

However, the Workflow Management community does not seem to realize the nature and size of the challenge they are now facing. The CSCW community may not have ready solutions to offer but we have, after all, an idea of the nature and size of the challenge.

7. REFERENCES

[1] T. Andrews. Business Process Execution Language for Web Services (BPEL4WS). Version 1.1. http://www-128.ibm.com/developerworks/library/specification/ws-bpel/, 2003.

[2] S. Brahe. BPM on top of SOA: Experiences from the Financial Industry. In P. D. G. Alonso and M. Rosemann, editors, *BPM2007*, volume 4714 of *LNCS*, pages 96–111. Springer, Heidelberg, 2007.

[3] P. H. Carstensen and C. Sørensen. From the social to the systematic: Mechanisms supporting coordination in design. *Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computing*, 5(4):387–413, 1996.

[4] M. Divitini and C. Simone. Supporting different dimensions of adaptability in workflow modeling. *Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computing*, 9(3-4), 2000.

[5] C. Dudley, L. Rieu, M. Smithson, T. Verma, and B. Braswell. *WebSphere Service Registry and Repository Handbook*. IBM Redbooks. http://www.ibm.com/redbooks, 2007.

[6] C. A. Ellis. Information Control Nets: A Mathematical Model of Office Information Flow. In *Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems*, pages 225–240, Boulder, CO, 1979. ACM Press.

[7] C. A. Ellis and M. Bernal. Officetalk-d: An experimental office information system. In *Proceedings of the ACM SIGOA conference on Office information system*, Philadelphia, Pennsylvannia, 1982.

[8] M. Klein, C. Dellarocas, and A. Bernstein. Adaptive Workflow Systems. *Special issue of Computer Supported Cooperative Work (CSCW): The Journal of Collaborative Computingb*, 9(3-4), 2000.

[9] M. Kloppmann, D. Koenig, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, and I. Trickovic. WS-BPEL Extension for People – BPEL4People. A Joint White Paper by IBM and SAP. ftp://www6.software.ibm.com/software/developer/library/ws-bpel4people.pdf. Last accessed May 2007, July 2005.

[10] T. Kreifelts. Experiences with the DOMINO office procedure system. In L. J. B. M. Robinson; and K. Schmidt, editors, *ECSCW'91: Proceedings of the Second European Conference on Computer-Supported Cooperative Work*, pages 117–130, Amsterdam, September 1991. Kluwer Academic Publishers, Dordrecht.

[11] T. Kreifelts, E. Hinrichs, and G. Woetzel. Sharing to-do lists with a distributed task manager. In G. D. Michelis, C. Simone, and K. Schmidt, editors, *ECSCW'93: Proceedings of the Third European Conference on Computer-Supported Cooperative Work*, pages 31–46. Kluwer Academic Publishers, Dordrecht, 1993.

[12] F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques*. Prentice Hall, 2000.

[13] K. Schmidt and C. Simone. Coordination Mechanisms: Towards a Conceptual Foundation of CSCW Systems Design. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 5:155–200, 1996.

[14] L. A. Suchman. Office procedures as practical action: Models of work and system design. *ACM Transactions on Office Information Systems*, 1(4):320–328, 1983.

[15] W. M. P. van der Aalst. The Application of Petri Nets for Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[16] W. M. P. van der Aalst, A. H. M. Hofstede, and M. Weske. Business Process Management: A Survey. In W. van der Aalst et al., editor, *BPM2003*, volume 2678 of *Lecture Notes in Computer Science*, pages 1–12, 2003.

[17] W. M. P. van der Aalst and K. M. van Hee. *Workflow Management: Models, Methods, and Systems*. The MIT Press, 2002.

[18] M. D. Zisman. *Representation, Specification and Automation of Office Procedures*. PhD thesis, University of Pennsylvania, Wharton School of Business, 1977.