

Process based E-Services

A. Lazcano and G. Alonso

Department of Computer Science
Swiss Federal Institute of Technology (ETHZ)
ETH Zentrum, CH-8092 Zürich, Switzerland
{lazcano, alonso}@inf.ethz.ch

Abstract. In this paper we describe WiseFlow, a software platform currently being used to provide a variety of electronic services within the maritime industry. WiseFlow lies at the core of an international trading community formed by certification societies, consulting firms, shipyards, steel mills, manufacturers, insurance companies and government agencies. As such, this community revolves around well defined business processes shared among all participants: certification of materials for the shipping industry. WiseFlow supports the daily execution of these processes and implements different business models from several of the participants.

1 Introduction

Maritime certification societies face a daunting inter-enterprise document handling problem. International law requires every part of a vessel, from manufactured components (e.g., engines) to raw materials (e.g, steel plates), to be certified. These certificates are essential to obtain navigation permits and play a crucial role in the insurance and demarcation of responsibilities in case of accidents. During the lifetime of large commercial vessels like oil tankers or container carriers, the number of certificates associated to the vessel can reach several hundred thousands. These certificates need to be maintained, updated, transferred, and consulted as the vessel undergoes repairs, changes ownership, changes its navigability ratings, changes insurance companies, etc. The certification of components and materials is carried out by *certification societies*, internationally recognized as trusted entities in charge of the issuing and custody of the certificates. Historically, certification societies have faced little competition as they tended to be national entities that concentrated in geographically well defined markets. This situation is rapidly changing and, as a result, there has been a mounting pressure to modernize the certification process as a whole. In addition, current practice is based on keeping several hard copies of the same certificate at several places: at the certification society, at the shipyard where the vessel was produced, at the ship owner, and even at the manufacturers of components and raw materials. This is clearly inefficient and opens up significant opportunities for third parties willing to provide electronic services.

In this regard, both the certification process and the handling of certificates are textbook examples of the type of inter-company data exchanges that could

greatly benefit from switching to an electronic format. The realities of the market, however, make it very hard to come up with solutions that are, simultaneously, cost efficient, usable by all those involved, and complying with existing business procedures. On the one hand, the certification process involves many different participants from different companies, all of them widely distributed geographically. On the other hand, not all participants are *equals* in terms of their ability to implement electronic processes and the rights they have with respect to the overall process. In addition, the process itself has different interpretations depending on the participant, making it very difficult to find the necessary level of consensus to get all those involved to adopt the same technology.

We have tried to address all these issues by deploying the WiseFlow system. WiseFlow is unique in that it not only addresses the typical technological problems associated with e-commerce activities but also covers the business needs of different participants in the certification process. Thus, WiseFlow can be alternatively or simultaneously used as a workflow engine, as an electronic commerce platform for implementing inter-enterprise business processes, and as a tool for providing web and XML based electronic services. This flexibility offers significant advantages, specially in an environment where, as pointed out above, different partners have different interpretations of the process at hand and need different types of support. By providing a unique tool to all users, WiseFlow has greatly helped to reach a consensus on the technological basis. In this, we agree with other authors who see process automation as a foundation for electronic commerce [CS00,CIJ⁺00].

The paper is organized as follows. In Section 2 we define the certification process and describe the different electronic services supported. Section 3 introduces the architecture of the system as well as the different configurations in which it has been deployed. Section 4 concludes the paper.

2 WiseFlow: requirements for e-services

2.1 The certification process

Figure 1 contains a simplified version of the certification process. The real process includes around 80 execution steps. The certification process is started by a customer who orders certified steel. This order is received at the *steel mill*, and analyzed to decide if it can be manufactured. Once the order has been confirmed, the steel mill can start planning its production. A certificate number for this material is issued by the *certification society* on request.

The steel mill tests the manufactured material and includes the results with some additional information in a report. This is quality data that has to be sent to the certification society, but not before the necessary certificate number has been received. The steel mill may generate the quality data either in Edifact or PDF format. In the case of Edifact format, the document is sent to the *consulting firm* for further processing. They apply a series of conversions to the data and make corrections in case of errors. The resulting XML document is then transferred to the certification society.

The certification society receives either a XML or a PDF document. The XML document is used to generate a certificate which is then archived and copies of it distributed among customer, consulting firm and steel mill. If the document received by the certification society is in PDF format, it is directly stored and copies of it are also distributed.

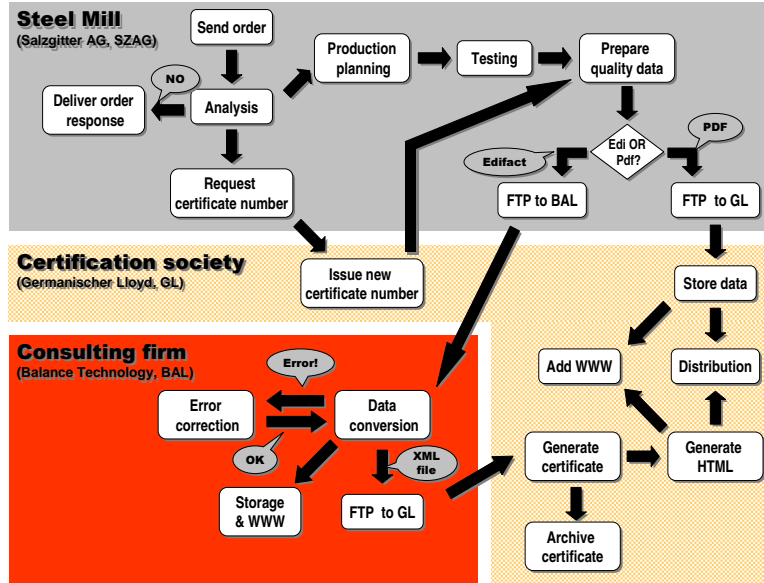


Fig. 1. A summary of the certification process used in the maritime industry

2.2 WiseFlow for a certification society

For the certification society, the primary interest is the automation of their (intra-organizational) certification subprocess using a workflow engine. This interest follows conventional wisdom regarding the advantages and benefits of automating business processes [CD00] [BGS⁺99], [MWGW99]. Closely related to the process automation is the problem of document management. Currently, the final certificates are stored in paper, which has both a high overhead and does not make economic sense any more. Automating the process using a workflow tool will immediately allow the electronic storage and management of the certificates, keeping track not only of the certificate itself but also of the process followed to create the certificate (nowadays this information is obtained by following the paper trail of the process).

In addition, certification societies see the automation of their internal certification process as the cornerstone of *trading communities* where they will play an even more central role in the industry and offer their current services in a far more efficient way.

2.3 WiseFlow for a provider of e-services

The current certification process leaves much room for improvement and offers many opportunities for organizing e-services around it. The two existing services are EDI transcription of documents (acting as a proxy for companies without resources to deal with electronic documents) and storing certificates (mainly at the certification society). WiseFlow has been used to implement these services more efficiently and to create several new ones.

For instance, any participant in the certification process is likely to have several simultaneous processes involving different subsets of participants (steel mills, e.g., often obtain several certificates for the same material, each certificate being issued by a different certification society). To deal with such cases, a trusted intermediary could provide the necessary process support by acting as a process hub in charge of process execution and monitoring but not involved in the process itself. As another example, the coordination between the transport of the material and the delivery of the certificates is a cause of concern. It is not unusual for raw steel to arrive at a shipyard only to sit there unused until the certificates arrive weeks later. An e-service capable of reporting the status of the certification process could greatly enhance the coordination among participants and significantly reduce cost. Naturally, this service can be best provided by the process hub which could charge a premium for accessing this information aside from charging for running the process itself.

Finally, handling the process history can also be an interesting e-service. For any given participant, the certification process contains key information about the way processes are executed. For instance, it can help a steel mill to identify inefficiencies in the delivery process. Or it can help a shipyard to plan their purchases in advance in order to implement a just-in-time operation.

2.4 WiseFlow for companies less committed to technology

In the certification process, not all the companies have the same computing resources. This is a very important practical limitation that needs to be kept in mind. The certification process usually involves companies in developing or even third world countries, where access to computer and modern networks is limited. Even in Europe, many companies are not willing to make the effort to adopt a large information system. Unfortunately, many of the current B2B (business to business) solutions proposed by industry assume that all participants have a sufficiently large amount of computing and communication resources. From our experience, such systems cannot be used in an environment as heterogeneous as the maritime industry. In fact, the biggest difficulty in implementing an inter-organizational business process are not the big players and compatibility problems between their computing equipment but the smaller players and their lack of computing resources.

This problem was recognized very early on in the design of WiseFlow. Accordingly, WiseFlow allows clients to be connected to the system using a minimal amount of software. All that is needed is a WiseFlow proxy (written in Java)

in the client machine. This simplifies the installation procedure as the code can be easily downloaded from a web site. Once the client is registered with the WiseFlow server, it can be used as one more node in the system. Any additional code can be produced on demand by downloading it from the server whenever needed.

3 WiseFlow: the system

WiseFlow has evolved from several systems built at ETHZ. The core of WiseFlow is OPERA [Hag99,AHST97], a process support kernel that provides the basic workflow engine functionality and a number of programming language extensions (e.g., event handling [HA99a], exception handling [HA00], spheres of atomicity [HA98], and high availability [HA99b]). OPERA was first extended to build WISE (Workflow based Internet Services) [AFH⁺99]. WISE supports trading communities that interact based on *virtual business processes* [LASS00]. WISE incorporates a commercial process design tool (IvyFrame [Ivy98,Lie98]) that has been redesigned to work both as a process definition interface and a monitoring tool. In the MariFlow project (an EU project), the WISE system became WiseFlow once it was adapted to the needs of the maritime certification process. This adaptation includes logging capabilities, more sophisticated interfaces and API for handling documents, the ability to work across corporate firewalls, security, and the ability to arbitrarily combine WiseFlow servers in different locations to form a meta-server. These features are currently being extended as part of the INVENT project (Infrastructure for Virtual ENterprises).

3.1 Architecture of WiseFlow

A WiseFlow server can be seen as a modular workflow engine that can recursively call other WiseFlow servers to execute subprocesses. WiseFlow is organized around three service layers (Figure 2.b): *database services*, *process services* and *interface services*.

The database service layer acts as the storage manager. It encompasses the storage layer (the actual databases used as repositories) and the database abstraction layer (which makes the rest of the system database independent). The storage layer is divided into five *spaces*: template, instance, object, history, and configuration, each of them dedicated to a different type of system data. Templates contain the structure of the processes. When a process is to be executed, a copy of the corresponding template is made and placed in the instance space. This copy is used to record the process' state as execution proceeds. For each running instance of a process the instance space contains a copy of the corresponding template. Storing instances persistently guarantees forward recoverability, i.e., execution can be resumed as soon as the failure is repaired, which solves the problem of dealing with failures of long lived processes [ST96,DHL91]. In addition, the instance space is the basis for e-services based on notification of the status of a process. Objects are used to store information about externally

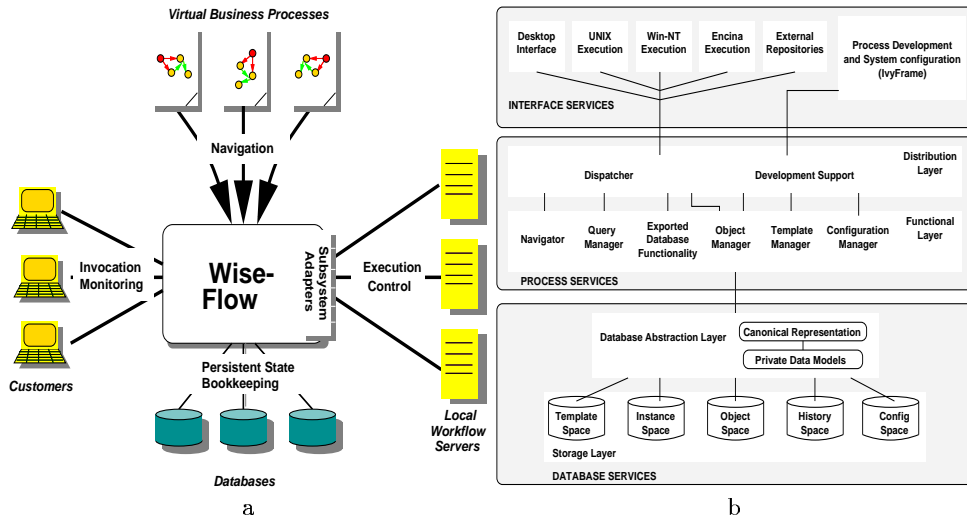


Fig. 2. a) General architecture of WiseFlow, b) Internal organization of the engine

defined data. They allow WiseFlow to interact with external applications by acting as a proxy containing the information indicating how to access external data [BRS96]. The history space is used to store information about already executed instances (as in, for instance, [PMW99]). It contains a detailed record of all the events that have taken place during the execution of processes, including already terminated processes. Currently we use Oracle for this purpose as the history space constitutes the basis for several e-services. Finally, the configuration space is used to record system related information such as configuration, access permissions, registered users, Internet addresses, program locations, and so forth. The database abstraction layer implements the mechanisms necessary to make the system database independent.

The process service layer contains all the components required for coordinating and monitoring the execution of processes. The most relevant components for the purposes of this paper are the *dispatcher* and *navigator* modules. The dispatcher deals with physical distribution and acts as resource allocator for process execution. It determines in which node the next step will execute, locates suitable nodes, checks the site's availability, performs load balancing, and manages the communication with remote system components. The navigator acts as the overall scheduler: it "navigates" through the process description stored in the main memory, establishing what to execute next, what needs to be delayed, and so forth. Once the navigator decides which step(s) to execute, the information is passed to the *dispatcher* which, in turn, schedules the task and associates it with a processing node in the cluster and a particular application. If the choice of assignment is not unique, the node is determined by the scheduling and load balancing policy in use. The dispatcher then contacts the *program execution client* (PEC); this is a small software component present at each node respon-

sible for running application programs on behalf of the WiseFlow server. The PEC is responsible for invoking the application program that will complete the computational step. It is written in Java and, thus, it is platform independent, allowing WiseFlow to work with heterogeneous nodes. The PEC contains all the mechanisms that allow WiseFlow to interact with applications in different hardware and software platforms. The small size of the PEC allows providers of e-services to easily attach remote nodes (in the companies using the e-services) to their own WiseFlow server for those cases where user input maybe required. Users interact with the system via *desktop interfaces*, which are also used to inform the user of any activity that they need to execute as part of a process (similar to worklists in workflow engines).

Processes are created using a commercial tool, *IvyFrame* [Ivy98,Lie98], which is internally based on Petri-nets and supports not only the modeling of business processes but also sophisticated analysis of its behavior (bottlenecks, average execution times, costs, delays, what if analysis, etc.). Using the process definition tool, it is possible to perform *process creation* and *configuration management*. The configuration management allows users to specify the hardware and software characteristics of the computer infrastructure to be used in the execution of a process (IP addresses, type of OS, CPU specifications, etc.).

WiseFlow uses internally a language called *Opera Canonical Representation* (OCR) [Hag99] to describe processes. The graphical representation is compiled into an OCR text file that is used to create process templates. In OCR, a process consists of a set of *tasks* and a set of *data objects*. Tasks can be *activities*, *blocks*, or *subprocesses*. The data objects store the input and output data for the tasks and are used to pass information around a process. More details of OCR and the internal data structures of OPERA can be found in [Hag99].

3.2 Additional functionality: High availability

Since the state of a process instance is persistent, WiseFlow can resume the execution of a process at the point where it was left off when a failure occurred. However, in certain cases it should even be possible to resume execution before the failure is repaired by using a reliable backup strategy. WiseFlow provides such a mechanism [HA99b], implemented using application semantics to optimize the exchange of information between the primary copy and the backup. There is no dedicated backup, and instead databases will act as both primaries and backups depending on the particular process instance. This backup mechanism is also used to dynamically migrate processes as they execute from one server to another. Such functionality is of great help to balance the load on a large system.

3.3 Deployment options

WiseFlow can be used as stand alone engine or as a cluster of engines over a wide area network. In stand alone mode, it works as a workflow engine capable of controlling remote (across the Internet) clients. From the point of view of

an e-services provider, this capability opens up interesting opportunities as it allows designers to link up several companies into a single process deploying only the PEC. Communication across and through firewalls is achieved through a proprietary communication protocol implemented in WiseFlow. The protocol offers a variety of possibilities, including RMI tunneling through HTTP to get through firewalls without compromising security. The protocol also supports encryption, digital signatures and authentication so that system designers can implement the level of security appropriate to each case.

As a cluster of WiseFlow servers, the system allows the implementation of trading communities through a network of interrelated business processes that are combined to form a virtual business process. The system is based on a number of WiseFlow servers acting as clients of each other. Needless to say that controlling this execution has a significant potential as an e-service. Using this approach, the e-services provider runs a skeleton of the certification process activating subprocesses or activities at different companies as needed. The provider does not need to know the details of these subprocesses (which is proprietary information in most cases), it only needs to know where they fit within the overall process (which is known to all participants). The same protocol mentioned above is here used to allow communication across servers located behind firewalls. The provider can therefore act as both a coordinator for the overall process and a proxy for companies that cannot run electronic processes.

3.4 Data storage, notification

Keeping track of the execution of a process lies at the center of several e-services around the certification process. Typically, there are many activities that depend on the status of the process and it is a great advantage to be automatically notified when the process has reached a given state. For instance, the steel is sent to the shipyard only when a certificate has been issued, or production does not start until all certificates for a batch of steel have arrived. Such dependencies can be implemented in several ways [BGS⁺99]. If these are dependencies between processes, WiseFlow provides an event propagation mechanism that allows a WiseFlow server to send events to other servers that have subscribed to those events [HA99a]. Since a WiseFlow process can raise events at any point in time, this allows designers to interconnect different business processes. In addition, to expand the number of potential users for such a notification service, WiseFlow combines its instance space with trigger functionality. The triggers are used to implement different notification services (web pages, e-mail messages) according to each user.

WiseFlow also provides an interface that allows to monitor in real time the progress of a given process. This interface is based on the same application as the process definition tool. The novelty of the approach used in WiseFlow is that this interface is entirely written in Java and can be downloaded as a Java applet, thereby allowing the remote monitoring of a process. This functionality is crucial to companies that outsource the execution of their part of the certification

process. In spite of not owning the server and not driving the execution, such companies can keep track of their processes by using the WiseFlow interface.

3.5 Performance and dependability

After extensive testing, the WiseFlow system has proven to meet the requirements of the maritime community. The most important of these requisites is that the system should be able to support 3000 certifications per month. WiseFlow has been shown to tolerate four times that load using a single server.

To increase the scalability of the system one needs only additional WiseFlow servers and implement the certification process as a skeleton that invokes subprocesses at the different sites. Since load is distributed among the servers, this allows the system to support many more process instances. The e-services provider could coordinate the overall process and offer process monitoring, as well as process history analysis as an interesting e-service. Among the requirements stated at the beginning of this project were a list of typical queries: What is the status of a particular certificate, what is the average processing time of a particular step, list all my certificates, list all the processes of last year, etc. Our current process history based on Oracle supports this and can provide all this important information related to process execution.

4 Conclusions

The maritime industry is one of the many business domains where the different companies involved can benefit enormously from the modernization of their shared business processes. The specific characteristics of this environment offer also a good opportunity to all those interested in providing electronic services to participants in the trading community. In this paper we have presented WiseFlow, a system that can be used in this context as a platform for implementing intra- and inter-enterprise business processes, as well as a tool for providing e-services.

References

- [AFH⁺99] G. Alonso, U. Fiedler, C. Hagen, A. Lazcano, H. Schuldt, and N. Weiler. Wise: Business to Business E-Commerce. In *Proceedings of the IEEE 9th International Workshop on Research Issues on Data Engineering. INFORMATION TECHNOLOGY FOR VIRTUAL ENTERPRISES (RIDE-VE'99)*. Sydney, Australia, March 23-24, 1999., March 1999.
- [AHST97] G. Alonso, C. Hagen, H.J. Schek, and M. Tresch. Distributed Processing over Stand-alone Systems and Applications. In *Proceedings of the 23rd International Conference on Very Large Databases (VLDB'97)*, Athens, Greece, August 1997.
- [BGS⁺99] D. Baker, D. Georgakopoulos, H. Schuster, A. Cassandra, and A. Cichocki. Providing customized process and situation awareness in the collaboration management infrastructure. In *Proceedings of the Fourth International Conference on Cooperative Information Systems*, 1999.

- [BRS96] Stephen Blott, Lukas Rely, and Hans-Jörg Schek. An open abstract-object storage system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, June 1996.
- [CD00] Fabio Casati and Angela Discenza. Supporting workflow cooperation within and across organizations. In *Proceedings of the 2000 ACM Symposium on Applied Computing*, Villa Olmo, Como, Italy, March 2000.
- [CIJ⁺00] Fabio Casati, Ski Ilnicki, LiJie Jin, Vasudev Krishnamoorthy, and Ming-Chien Shan. Adaptive and dynamic service composition in eflow. In *12th International Conference CAiSE 2000*, Stockholm, Sweden, June 2000.
- [CS00] Fabio Casati and Ming-Chien Shan. Process automation as the foundation for e-commerce. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, Cairo, Egypt, September 2000.
- [DHL91] U. Dayal, M. Hsu, and R. Ladin. A Transaction Model for Long-running Activities. In *Proceedings of the Sixteenth International Conference on Very Large Databases*, pages 113–122, August 1991.
- [HA98] C. Hagen and G. Alonso. Flexible exception handling in the OPERA process support system. In *Proc. of the 18th Intl. Conference on Distributed Computing Systems*, Amsterdam, The Netherlands, May 1998.
- [HA99a] C. Hagen and G. Alonso. Beyond the black box: Event-based inter-process communication in process support systems. In *Proc. of the 19th Intl. Conference on Distributed Computing Systems*, Austin, Texas, USA, May 1999.
- [HA99b] C. Hagen and G. Alonso. Highly Available Process Support Systems: Implementing Backup Mechanisms. In *18th IEEE Symposium on Reliable Distributed Systems*, Lausanne, Switzerland, October 1999.
- [HA00] C. Hagen and G. Alonso. Exception Handling in Workflow Management Systems. *IEEE Transaction on Software Engineering*, 26(9), September 2000.
- [Hag99] C. Hagen. *A Generic Kernel for Reliable Process Support*. Ph.D. Dissertation, ETH Nr. 13114, 1999.
- [Ivy98] IvyTeam. Structware'98 Process Manager. Available through <http://www.ivyteam.com>, 1998.
- [LASS00] A. Lazcano, G. Alonso, H. Schuldt, and C. Schuler. The WISE approach to Electronic Commerce. *International Journal of Computer Systems Science & Engineering*, September 2000.
- [Lie98] H. Lienhard. IvyBeans - Bridge to VSH and the project WISE. In *Proceedings of the Conference of the Swiss Priority Programme Information and Communication Structures, Zürich, Switzerland*, July 1998.
- [MWGW99] Peter Muth, Jeanine Weißenfels, Michael Gillmann, and Gerhard Weikum. Integrating light-weight workflow management systems within existing business environments. In *Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Australia*, pages 286–293. IEEE Computer Society, 1999.
- [PMW99] Michael Gillmann Peter Muth, Jeanine Weissenfels and Gerhard Weikum. Workflow history management in virtual enterprises using a light-weight workflow management system. In *Proceedings of the Ninth International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises*, Sydney, Australia, March 1999.
- [ST96] B. Salzberg and D. Tombroff. DSDT: Durable Scripts Containing Database Transactions. In *Proceedings of the 12th International Conference on Data Engineering*, New Orleans, Louisiana, USA, February 1996.