AFRICA: Workflow Interoperability based on XML-messages

Michael zur Muehlen, Florian Klein

University of Muenster
Department of Information Systems
Steinfurter Str. 109
48149 Muenster, Germany
{ismizu|isflkl}@wi.uni-muenster.de

Electronic business relationships over the internet, such as the linking of cross-organizational business processes in virtual supply chains and other scenarios enjoy an increasing interest in the business world. Especially the use of XML as a domain-independent encoding standard for business documents has led to the development of standardized business frameworks, and the development of interoperability mechanisms that support these standards is underway. In this paper we describe an architecture for the support of cross-organizational workflows through XML messages. This architecture has been implemented and tested within the AFRICA[1] project at the University of Muenster, Germany. While our work is based upon the emerging Wf-XML standard of the Workflow Management Coalition, it contains a number of significant enhancements that provide a secure, reliable management of global workflow processes.

1 Motivation for Interorganizational Workflow Management

The need for companies to expand the automated enactment of their business processes beyond the boundaries of their own organization is driven by the resulting savings in transmission time, gains in data quality and improved monitoring capabilities about processes at the site of business partners. The current movement towards electronic data interchange is fueled by the relatively inexpensive exchange of business data over the Internet using data encoded in the eXtensible Markup Language (XML) [1], which fosters the development of vendor-independent frameworks that aim to standardize data schemas for common business documents, such as purchase orders, delivery notes, invoices, etc.

Workflow management technology supports the execution of business processes through the automated coordination of tasks, data, application logic and workflow participants. While the use of workflow technology within single organizations is a well understood concept, and the number of implementations is increasing [2], the use of workflow management for the coordination of cross-enterprise processes is still at a very early stage. Currently, workflow management systems are shifting from stand-alone applications towards embedded solutions, that are delivered as inherent components of surrounding application systems, such as enterprise resource planning systems [3]. As a consequence, many organizations that currently have no workflow management system in place will have the option of automating part of their business processes using these embedded workflow applications without the necessity to purchase a separate workflow management system. In order to enable these companies to participate in interorganizational workflow settings, we have developed a prototype, that enhances existing workflow management systems with the capability to interoperate through standardized XML messages. This system has been successfully tested in a helpdesk scenario. After a discussion of related standards in the following section, we discuss the design and implementation of the AFRICA prototype in section 3. A reference to related research projects and a résumé and outlook to future developments conclude this paper.

2 Workflow Interoperability and XML Frameworks

2.1 Workflow Interoperability

In the area of workflow interoperability, a number of standardization efforts have been in place, namely by standardization organizations like the Workflow Management Coalition, the Object Management Group and the Internet Engineering Task Force.

2.1.1 Workflow Management Coalition Interface 4

The reference model of the Workflow Management Coalition (WfMC) identifies five functional interfaces, that connect a workflow management system with external application systems, of which Interface 4 deals with workflow interoperability [4]. The interface definition consists of an abstract specification of those API calls,

[1] A Flexible, Reliable, Intelligent Communication Architecture

that can be used to instantiate a workflow on a remote workflow management system, to change the execution status of a workflow instance or to query the data processed in a remote workflow instance [5]. Instantiations of the abstract Interface 4 specification that relate to a specific message encoding scheme (e. g. HTML, MIME, etc.) are called bindings. So far, only a MIME binding has been published by the WfMC [6]. The applicability of this specification has been demonstrated in an interoperability challenge [7] that was carried out in March 1999.

2.1.2 Object Management Group Workflow Facility

The Object Management Group (OMG) has standardized a facility that provides workflow services through an object request broker [8]. The Workflow Facility describes a set of workflow execution objects and their respective interfaces that can be used for workflow interaction in business object environments. The Workflow Facility standard has been officially released by the OMG, but adjacent components are still awaiting standardization, such as a resource assignment interface for the association of workflow participants to workflow activities [9] and a process modeling standard for the design of workflow processes [10].

2.1.3 Simple Workflow Access Protocol

The Simple Workflow Access Protocol (SWAP) was created in 1998 through an industry consortium under the auspices of Netscape, Oracle, SUN [11, 12]. The SWAP-specification, which is still in draft status, can be used for the control of a workflow instance through a remote control instance by exchanging XML messages through HTTP. The original industry consortium has handed over the SWAP specification to the Internet Engineering Task Force (IETF) for standardization. However, progress has been slow with the standardization of SWAP. Since the pressure for an interoperability standard using XML is mounting, the Workflow Management Coalition has adopted the basic ideas of SWAP and merged them with an XML binding of the WfMC Interface 4 specification. The result is the forthcoming Wf-XML standard [13], which is described in the next section.

2.1.4 Wf-XML

Wf-XML [13, 14] is an interoperability standard (currently in beta-status), defined by a WfMC working group, that combines the elementary concept of SWAP with the abstract commands defined by the WfMC Interface 4. Wf-XML defines a set of request/response messages that are exchanged between an observer (which may or may not be a workflow management system) and a workflow management systems that control the execution of a remote workflow instance:

? `CreateProcessInstance` instantiates a new workflow instance within a remote workflow management system.

? `ChangeProcessInstanceState` manipulates a workflow instance on a remote system (starting, suspending, terminating the remote instance etc.).

? `GetProcessInstanceData` requests the status of the remote workflow instance.

? `ProcessInstanceStateChanged` signals to the requesting party (i. e. the observer of the workflow process) that the remote process has been completed and passes the result data to the requesting party.

Due to the broad support by workflow vendors and the proximity of standardization, Wf-XML was chosen as the message format for the AFRICA prototype.

2.2 XML Business Frameworks

Recently a number of standardization organizations for XML business frameworks have appeared on the Internet. The most well-kown organizations of this kind are BizTalk and RosettaNet[2].

The *BizTalk* forum was created by Microsoft and aims at the definition of guidelines for the publication of XML schemas by independent vendors [15]. The use of XML messages for the integration of software systems is propagated by the Simple Object Access Protocol, which is an XML/HTTP-based protocol for the platform-independent access to services, objects and servers over the Internet.

The *RosettaNet* consortium was formed by various manufacturers and suppliers of hard- and software in order to standardize supply chain processes within the IT industry domain [16]. The RosettaNet specification defines Partner Interface Processes for various supply chain processes such as management of purchase orders, product and technical data interchange, and order status handling.

[2] Besides BizTalk and RosettaNet a number of other standardization organizations for XML frameworks exist, such as Open/EDI, OBI, CommerceNet and the Open Trading Protocol Consortium.

3 The AFRICA Prototype

The AFRICA project was initiated in October 1999 with the aim of building a reliable infrastructure for business-to-business workflows, using XML for message encoding. The focus was on incorporating complex, non-sequential process models involving multiple partners and the integration of a global monitoring service. Since the project team had a number of commercial workflow management systems available for testing and integration purposes, it was decided not to implement a workflow engine with interoperability features, but instead an add-on component (wrapper), that can be added to existing workflow installations. A reference implementation of this wrapper was finished by March 2000, clearly demonstrating the potential of XML-based process communication.[3]

3.1 Design Rationale

For the design of the AFRICA prototype, a number of design principles were employed, to make the system usable in a large number of contexts, namely system independence, reusability, security and support for n-party processes.

System independence: The AFRICA prototype should enable companies to participate in cross-organizational workflows without modifying existing workflow management systems. For this reason, the prototype was implemented as a wrapper that sits on top of an existing application system and encapsulates the Wf-XML message handling from the underlying system. Vendor supplied APIs are used to access the respective systems, leaving the system integrity untouched.

Reusability: The AFRICA prototype was designed with the goal of using as much of the system code as possible in different environments. Therefore, a three tier architecture was developed, separating the transport layer, process logic layer and abstraction layer. For changing transport protocols or security mechanisms, only the transport layer needs to be adjusted, while during the migration to a different workflow management system only the abstraction layer is changed accordingly, leaving the transport and process layer intact.

Security: Communication between two AFRICA-enabled systems should be secure and reliable. In order to achieve this, additional information have been inserted in the transport section of the Wf-XML messages that are evaluated by the transport layer of the wrapper.

Support for n-party processes: While most interoperability frameworks focus on the peer-to-peer interaction between two business partners, the goal of the AFRICA project was the support for an arbitrary number of involved parties (e. g. a complex supply chain with several involved parties). In order to maintain the overall consistency of the process as well as to provide monitoring information about a global process regardless of the local enactment, the notion of a GlobalProcessID was added to the Wf-XML messages.

3.2 Message Format

The AFRICA wrapper uses an extended Wf-XML format for the exchange of messages. Each message consists of the four parts Transport, Security, Header and Body.

```
<?xml version="1.0"?>
<WfMessage>
   <WfTransport>
      ...
   </WfTransport>
   <WfSecurity>
      ...
   </WfSecurity>
   <WfMessageHeader>
   ...
   </WfMessageHeader>
   <WfMessageBody>
      ...
   </WfMessageBody>
</WfMessage>
```

Figure 1. Overall message structure

The `WfTransport` section groups those elements that are relevant at transportation time, before the message reaches its eventual recipient, i.e. a local process instance, e. g. the sender and recipient of the message as well as a correlation key for the identification of request/response pairs.

The `WfSecurity` section contains a unique identifier for each message and a timestamp. This information is used to identify lost, obsolete or intercepted messages and to acknowledge the receipt of the message by the transport layer (see section 3.3).

The `WfMessageHeader` section contains – different from the original Wf-XML standard – the global process identifier that this message relates to. Each wrapper only needs to know the mapping between the global process and its own local process instances, but does not need to keep track of the local naming schemas of other involved parties. The header section also contains the identifier of the operation to be executed in order to enable the preprocessing of this information.

The `WfMessageBody` section contains the details about the operation to be executed as well as the context data, i. e. the data that gets passed to the local workflow management system for further processing. We assume

[3] See http://pcwi501.uni-muenster.de/africa/index.html

no predefined structure of the context data, this way, data schemas that have been standardized by other organizations (cf. section 2.2) can be inserted here.

In addition to the operations defined by the Wf-XML standard, a number of additional operations were introduced, in order to facilitate global process management and the handling of monitoring information. These operations are PassProcessInstance, GetHistory and Notify.

`PassProcessInstance` hands the control of the global process over from one party to the next. The sender switches into the state “suspended” and is re-activated, when his wrapper receives a PassProcessInstance command himself. If a local process instance exists, the wrapper sets the state of this instance to active.running. If no local instance for the global process exists, the wrapper instantiates a new local instance, starts it and updates its lookup table with the local process ID. Note that the suspend and resume function only affects the state machine of the Africa wrapper. If the local workflow instance contains multiple threads (i. e. multiple concurrent paths within the workflow model), the suspend does not prohibit the continuous execution of these threads.

`GetHistory` requests monitoring information from a remote party. If the remote party has passed the process control to several other parties, the command is cascaded until the currently active party returns information about its current process status. The parties located in the middle between the sender and the final recipient of the command add their own process status information and pass a merged set of data on until it finally reaches the sender of the GetHistory command.

`Notify` actively “pushes” process status information to the observer of a remote process instance. If a remote party sees the necessity to inform the observer about certain events, the Notify command is used.

3.3 Technical Architecture

In order to facilitate reuse and encapsulation of information from lower levels, the AFRICA wrapper was designed using a three-tier architecture. Figure 2 illustrates the architecture of the system, using an incoming Wf-XML message as an example.

The *transport layer* handles the reliable and secure transfer of Wf-XML messages between AFRICA-enabled systems. The prototype uses TCP/IP port-to-port communication and SSL encryption for security measures. For other transport protocols or security mechanisms the transport layer has to be changed accordingly, but the process logic and abstraction layer remain untouched. When a message is received by the TCP Socket Listener, it is forwarded to the Open SSL Decryption unit, where the message is decoded. The plain-text XML message is then handed over to the security validation mechanism, that evaluates the timestamp and the security token of the message. The security token is sent back to the originator of the message to acknowledge the receipt. The wrapper keeps a backlog of the messages received during the last 20 minutes, older messages and messages with a duplicate security token are discarded as obsolete. The XML message is then sent to the process logic layer for further processing.

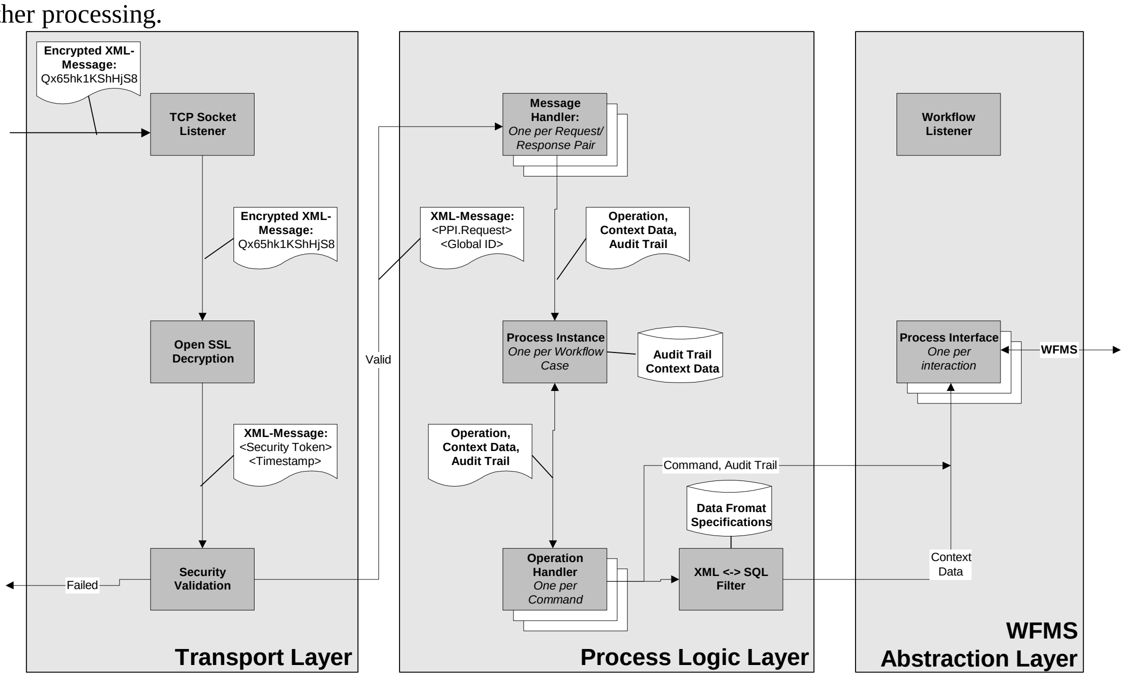


Figure 2. Three tier architecture of the AFRICA prototype (shown for an incoming message)

Within the *process logic layer*, the XML message is parsed, the Wf-XML command is separated from the context data of the message and a standardized call of abstraction layer API functions is issued. After the receipt of the message from the transport layer, the Message Handler component determines whether the message is a response to a request issued earlier (then the appropriate instance of the Message Handler is identified) or if the message is a request itself (in which case a new Message Handler is instantiated). The Message Handler extracts the Wf-XML command from the message and sends it with the context data and audit trail information to the Process Instance handler. This component reads the GlobalProcessID of the message and matches it with the local process instance that exists within the local workflow management system. It translates the global command into a local instance specific command and passes it on to a new instance of the Operation Handler. The Operation Handler then transforms the context data into the format required by the local workflow schema and passes the command and the context data through an API call to the abstraction layer. Depending on the content of an interoperability contract, the semantic interpretation of the context data is carried out in this layer. The resulting context data XML tree can be transformed in various ways, matching the requirements of the application system behind the abstraction layer.

The *abstraction layer* encapsulates proprietary API calls to the workflow management system of a specific vendor and exhibits a standardized API to the process logic layer. For the prototype two abstraction layers were completed, one for interaction with a custom-made web server and one for the Business Workflow component of an SAP R/3 system. Within this scenario, the command and context data received from the process logic layer are translated into SAP-specific BAPI (Business Object API) calls. In order to accommodate workflow management systems of different vendors, only the abstraction layer needs to be changed, leaving process logic layer and transport layer untouched.[4]

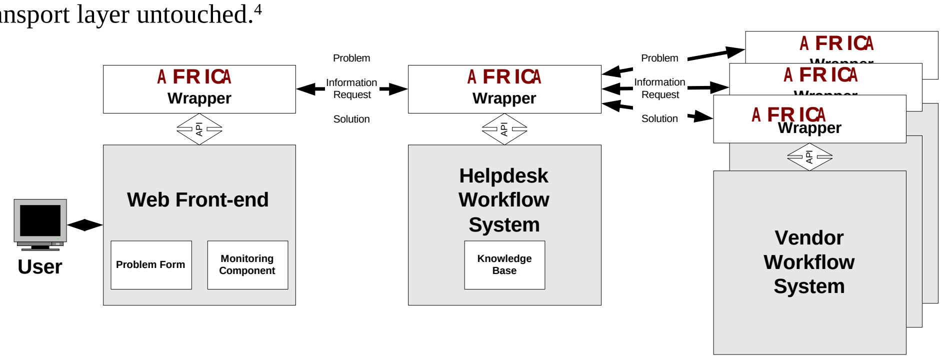
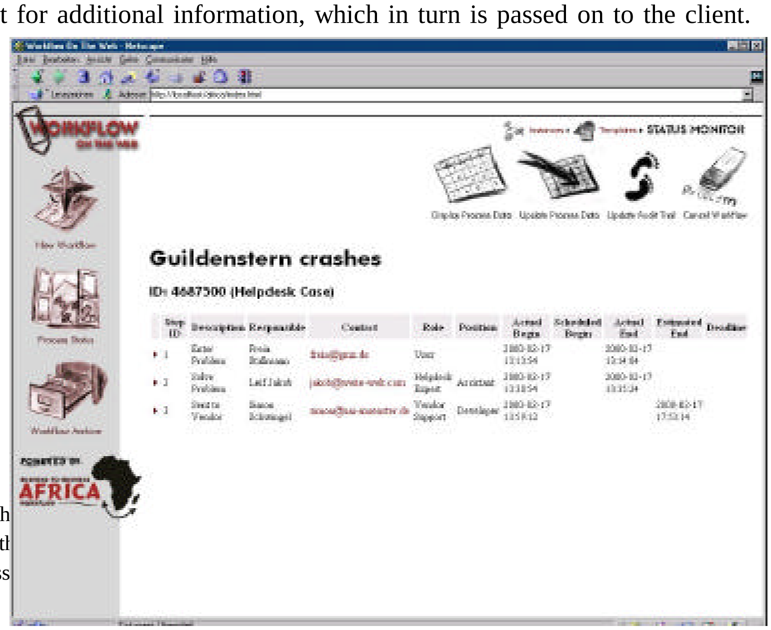


Figure 3. Helpdesk scenario

The AFRICA prototype was implemented using a 2-stage helpdesk scenario as an example. Within this scenario, a client has contracted an external service provider to perform helpdesk tasks, in case an employee of the client encounters soft- or hardware-related problems. The helpdesk then tries to solve the problem using his internal knowledge base. He may choose to either escalate the problem to a soft- or hardware vendors, send a solution to the client or request additional information from the client. The vendors in turn have the option of either sending a solution back or a request for additional information, which in turn is passed on to the client. Figure 3 gives an overview of the implementation scenario.

The client application of the helpdesk scenario can be used using only a web browser. Data entry and requests are handled via database-driven web pages. The interaction between the web front-end and the AFRICA wrapper is handled via server-side API calls, transparent for the user. Figure 4 shows the monitoring of a remote workflow instance through the web interface. After the problem was entered using a HTML form, the data was then sent to the helpdesk. The



[4] The process of sending out a Wf-XML message th workflow system calls out to the abstraction layer, tl Wf-XML message and integrates the GlobalProcess it to the appropriate recipient.

responsible workflow participant at the helpdesk site was unable to solve the problem and sent the problem description to the vendor site, where a developer is working on a solution and has given an estimate of the remaining processing time. From this screen, the user can review or update the problem description, actively request the audit trail of the process or cancel the workflow instance, which leads to a cascaded termination of workflow instances, that have been created at vendor sites by request from the helpdesk.

4 Related Research Projects

The *Interworkflow Project* at the Kanagawa Institute of Technology, Japan, focuses on the definition of a global workflow model for an interorganizational business process [17, 18]. This global workflow model defines the basic interaction between the parties involved and is then transferred into the workflow management systems of the parties involved for further refinement. During the enactment of the interorganizational workflow, both parties use the WfMC Interface 4 MIME binding for communication.

The *CrossFlow* project [19] is dealing with contract-based workflow interoperability between business partners. In this project, business relationships between a customer and a provider of services are modeled using contracts. The project uses an insurance and a logistics scenario for the demonstration of system interoperability.

WISE (Workflow-based Internet Services) is a project conducted at the Swiss Federal Institute of Technology [20, 21]. The aim of WISE is the development of an infrastructure for the support of cross-enterprise business processes in a virtual environment. The infrastructure of the WISE project includes the development of an internet workflow engine, which is beyond the scope of our approach. AFRICA focuses on a particular aspect of the WISE scenario, namely the interaction points of different workflow participants.

The *FlowJet* prototype [22], which is being developed at HP Labs, aims (among other goals) at the scenario-specific provision of application features. It can be configured as a stand-alone workflow application, a user agent, a personal work manager or a web service manager. It provides a service-exchange template for the interaction with external services, including data mapping, data binding and foreign process execution. However, no information is given as to how this communication is implemented in the current FlowJet prototype [23].

5 Summary and Outlook

In this paper we have presented the AFRICA framework for business-to-business workflow applications. It combines a standardized, extensible message format with a flexible and adaptable technical architecture. The applicability of this framework has been demonstrated by the implementation of a prototype scenario using commercial workflow management systems and a custom-made web-front-end. The extensions to the existing Wf-XML framework have been submitted to the WfMC for consideration in the next version of the Wf-XML standard. In the future our framework will be extended to accommodate a number of different workflow products. We are currently investigating the automated negotiation of communication parameters such as security mechanisms and protocol standards between AFRICA-enabled workflow systems as well as the automatic mapping of XML context data schemes into the proprietary format of the underlying workflow management systems.

Acknowledgements

The authors would like to thank the members of the AFRICA project team, Leif Jakob, Lutz Michaelsen and Simon Schwingel for their effort. Furthermore, we would like to thank the members of the WfMC Working Group 4, especially Rainer Weber and Keith Swenson for many useful comments on the AFRICA XML message format. Finally, we would like to thank the anonymous referees for their comments, which greatly helped to improve the quality of this paper.

References

[1] Walsh, Norman: A Technical Introduction to XML. http://www.xml.com/xml/pub/98/10/guide0.html.
[2] Fischer, Layna: Excellence in Practice Vol. III: Innovation & Excellence in Workflow Process and Knowledge Management, Lighthouse Point 2000.
[3] zur Muehlen, Michael; Allen, Rob: Autonomous and Embedded Workflow Management. A Workflow Management Coalition Classification. White Paper, Lighthouse Point, March 10, 2000. (available from http://www.wfmc.org)
[4] Workflow Management Coalition: Reference Model. Document Number WFMC-TC-1003, Brussels, 1995.
[5] Workflow Management Coalition: Workflow Standard Interoperability – Abstract Specification. Document Number WFMC-TC-1012. Version 2.0b. Winchester, 1999.
[6] Workflow Management Coalition: Workflow Standard Interoperability – Internet e-mail MIME Binding. Document Number WFMC-TC-1018. Version 2.0b. Winchester, 1999.
[7] Workflow Management Coalition: Interoperability Proving Framework. Document Number WFMC-TC-1021. Version 2.0b. Winchester, 1999.

[8] Object Management Group: Workflow Management Facility, Document Number bom/99-03-01. Framingham 1999. ftp://ftp.omg.org/pub/docs/bom/99-03-01.pdf

[9] Object Management Group: Resource Assignment Interface – Request for Proposals. Document Number bom/00-02-19. Framingham 2000.

[10] Object Management Group: Workflow Process Definition – Request for Proposal. Document Number bom/00-03-NN. Framingham 2000.

[11] Swenson, Keith: Simple Workflow Access Protocol (SWAP) Internet Draft (work in progress). (available from http://www.ics.uci.edu/~ietfswap/SWAP9807.html.

[12] Bolcer, Gregory Alan; Kaiser, Gail E.: Leveraging the Web To Manage Workflow. Collaborative Work: SWAP. IEEE Internet Computing, 3 (1999) 1, pp. 85-88.

[13] Workflow Management Coalition: Workflow Standard Interoperability – Wf-XML Binding. Document Number WFMC-TC-1023. Draft 0.9.2. (available from http://www.wfmc.org)

[14] Hayes, Jim; Perovian, Effat; Sarin, Sunil; Schmidt, Marc-Thomas; Swenson, Keith; Weber, Rainer: Wf-XML: Standards-based workflow interoperability for the Internet. To appear in: IEEE Internet Computing 4 (2000) 2, April 2000.

[15] BizTalk. http://www.biztalk.org

[16] RosettaNet. http://www.rosettanet.org

[17] Hayami, Haruo: Short Report on the Development and Experimental Proof of an Interworkflow Management System. Presentation at the Workflow Management Coalition Meeting in Vienna, February 1999. (available from http://www.wfmc.org)

[18] Hayami, Haruo: Development and Experimental Proof of an Interworkflow Management System. Presentation at the Workflow Management Coalition Meeting in Tokyo, December 1999. (available from http://www.wfmc.org)

[19] The CrossFlow Project. http://www.crossflow.org

[20] Alonso, Gustavo; Fiedler, U.; Hagen, C.; Lazcano, A.; Schuldt, H.; Weiler, N.: WISE: Business-to-Business E-Commerce. 9th International Workshop on Research Issues on Data Engineering (RIDE-VE'99). Sydney, Australia, March 23-24, 1999.

[21] Alonso, Gustavo; Fiedler, U.; Hagen, C.; Lazcano, A.; Schuldt, H.; Weiler, N.: WISE: An Infrastructure for E-Commerce. Proceedings of the Informatik'99 Workshop "Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications". Paderborn, Germany, October 6, 1999.

[22] Shan, Ming-Chien; Davis, Jim; Du, Weimin; Huang, Ying: HP Workflow Research: Past, Present, and Future. North Atlantic Treaty Organization – Advanced Study Workshop, Istanbul, Turkey, August 20, 1997.

[23] Shan, Ming-Chien: FlowJet: Internet-based E-Service Process Management. In: Proceedings of International Process Technology Workshop. Villars de Lans, France, September 1-3, 1999.