Telecooperation through World Wide Workflows

Johann Eder, Herbert Groiss

Department of Informatics-Systems
University Klagenfurt
A-9020 Klagenfurt, Austria
{eder,herb}@ifi.uni-klu.ac.at

Abstract

Business processes do not start or end at the boundaries of organizations. To support the participation of external agents in a workflow, the communication between workflow systems, or of a workflow system with other external IT-systems we propose a communication architecture which mainly relies on WWW-technology and the exchange of structured documents (forms). We separate the definition of a process and the assignment of agents from the communication with this agent to provide greater flexibility. We propose the use of HTML and XML forms for communication since these documents are readable for humans and are easily transformed into database records.

1. Introduction

Workflow management systems are an efficient means to increase productivity and quality of business processes. On one hand they contribute to a better documentation of processes and provide a better overview of active processes and thus allow for a more efficient managenment of business processes and improved process reporting. On the other hand, they increase throughput of processes mostly due to reduced transport times and fewer media gaps with their associated expense for multiple input of the same data.

These advantages of workflow management often end at the organizations boundaries. From business process reengineering of intra-organizational processes we learned that huge improvements can be made at the interfaces of organizational units. Therefore, we can expect that extending workflow technology to inter-organizational processes has a high potential for optimization.

The Workflow Management Coalition (WfMC) defined some ways of cooperation of workflow systems in the interface 4 of their reference architecture [7]. This cooperation is mainly technical in the sense that they provide interfaces that workflow managemt systems may exchange information. We take a diffent viewpoint here. We distinguish three different kinds of cooperations:

1. workflow participant from outside: Examples for such participants are customers for e-commerce or citizens for government agencies. This form of cooperation allows users from outside of the organization, to start a workflow, participate in a workflow by performing tasks. The user interacts with the workflow systems through a usual workflow client.

2. planned cooperation: Here the workflow systems interacts with the workflow system of a partner company. The cooperation can be preplanned, protocols for interaction are negotiated. This is the case where the interface 4 definitions of the WfMC can be helpful.

3. ad-hoc cooperation: In this type of cooperation the workflow system interacts with other workflow systems or IT systems without preplanned protocol and interfaces. Examples for such a cooperation are requesting quotes from several different vendors, answering to bids, etc. This kind of cooperation demands the employment of standards and the ability to flexibly adjust the interfaces of a workflow system.

In this paper we present an architecture to support all three kinds of cooperations. The main concepts of this architecture are:

- Separation of process definition from the way of interaction. When specifying a business process no information about the agent of a task should be necessary. The assignment of internal and external agents should be independent of the form of communication. The possible communcation channels should be chosen then with the help of the organizational model.

- Reduction of the assumptions about cooperation partners as much as possible. We want to make as little assumptions about cooopeartion partners as possible. For external human participants we only assume that they have access to a web browser. For IT systems we only assume that they can send and receive forms.

- Employment of communication standards as much as possible. In particular, we rely mainly on Web-technology and the EDIFACT standard for structured documents.

- Integration of different kinds of interaction in a uniform way.

In this paper we discuss different forms of interaction and show the communication architecture to support them. We introduce a model for maintaining information about external workflow participants and collaboration partners. We present the process definition language and the form interface, and finally we draw some conclusions.

2. Interactions

The platform of choice for electronic interaction between enterprises is the Internet with the popular services e-mail, ftp and WWW. Especially the latter has become very popular for accessing information and most workflow vendors built interfaces of their products to the Web. The services of the Internet can be used to support all steps of a business process:

Advertising: The seller of goods or services can provide information on Web pages, or mail potential customers directly. Interested persons find the Web sites by using a search engine or following a link.

Initialization: After a customer has found a supplier (or vice versa) a business process can be initialized: the customer initializes the process by making a request (for quote, for information, ...) or placing an order. Using the Internet, this can be done via a HTTP [1] form submission, the customer interacts with an application on the Web server of the supplier and is guided for providing the necessary information for initializing a process.

Cooperation: After initialization the business process normally contains further interaction between the partners: gathering additional information, bargaining, arrange delivery and payment conditions, and payment.

Monitoring: Customers should have the possibility to monitor the forthcoming of the process (a popular example of such functionality is the possibility to track the state of Federal Express packets, http://www.fedex.com/track_it.html).

The workflow management systems must be able to support cooperation of the above types and with different kinds of partners at various levels of integration. The workflow participants can be distinguished in:

1. **The occasional user**: The typical partner in non business-to-business electronic commerce. We can not expect any special software installed on his client and have to rely on the minimal requirements, someone has to fulfill to participate on electronic commerce:
   - Internet access,
   - receive and send electronic mail,
   - Web browser with HTML forms.

   Examples for this type of customers are people buying things over the Internet or performing electronic banking. The introduction of encryption of information via secure socket layer (SSL) allows the usage of this form of interaction for confidential information.

   Allowing all types of interaction (initialization, collaboration, and monitoring) the wfms must have a HTML interface providing this functionality. The notification of users about new tasks or other changes in the process can be done via e-mail. Fig. 1 shows the basic blocks of this architecture.

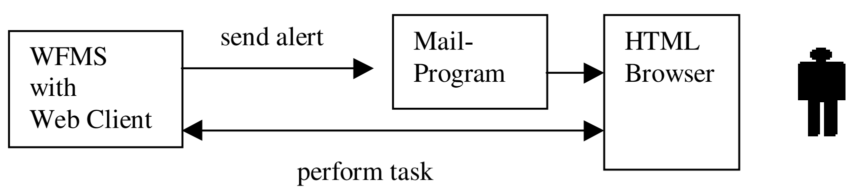

Fig. 1 Interaction with Web Client

   With the increasing power of standard Web browsers (HTML4, dynamic HTML, Java Applets) this simplest form of interaction gains increasing popularity in other interaction scenarios:

2. **Frequent users** of the system where it is not possible or feasible to install special-purpose software on the clients may use the Web Client. Consider for example doctors communicating with a hospital exchanging information about clients.

3. **Mobile users**: The third category of participants are employees from the organization connecting to the business process from the Internet, because they are on a business trip or teleworkers. The system architecture is the same again, the differences lay in the rights of the participants. From the customer, who is only allowed to see the business processes he is participating in (and from these only some restricted information, for example not all documents or not all other participants) to the manager on business trip who can monitor the processes of the whole department may work with the Web client.

   Providing a uniform user interface for people outside and inside the organization leads to great flexibility and reflects the structure of modern organizations where work is performed in highly decentralized networks of shifting project teams (this forms of organizations have been called *adhocracies,* [4]).

   One main advantage of this type of realization – no installation on the client – results in the main disadvantage: The client application has no access to local files or other resources. The whole interaction must therefore be done online. Signed Applets would provide a technical solution but the acceptance from the occasional user – giving away the right to write on the local disk - might be low.

4. The next category of users are **partners in other organizations**: Technically, they fulfill the same requirements as the above clients. The difference is, that they may have their own system installed, wanting to use one client for communicating with many other systems. The architecture is shown in Fig. 2.

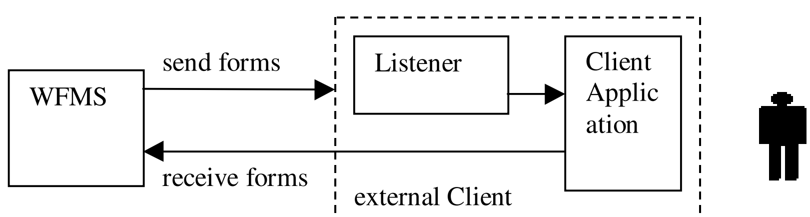

Fig. 2 Communication with external client

The presence of a special client allows downloading the information of the workflow and working offline. The communication is done via exchanging forms: when interaction is required the wfms sends the forms to the client, after performing the task the client sends back the forms to the wfms.

Monitoring running processes is also performed via exchanging forms, a request for information is sent to the wfms, which sends back the answer form. Additionaly, the monitoring functions of the Web client can be used.

5. The previous cases assumed a user communicating with a workflow system. Business-to-business e-commerce requires the communication of workflow systems. Fig. 3 shows the principal architecture, again using sending and receiving forms as communication mechanism.

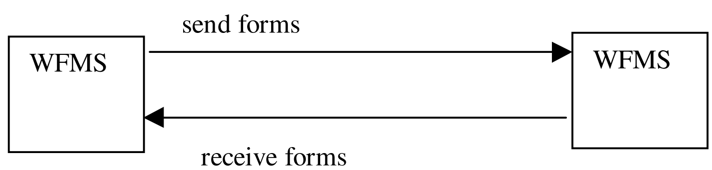

Fig. 3 Communication between workflow systems

Having to deal with this variety of clients and interfaces the challange is to provide a uniform architecture where the definition and execution of processes is done independently of the type of participants.

3. Maintaining organizations and people

A first step making the workflow system communicating with others is the representation of all participating agents in the workflow system. The following entities are maintained:

- *Roles*: Agents have roles, representing the type of job (secretary, manager, ...) or some skills (Java-programmer). Using roles as agents in process definitions make these definitions independent of particular persons.

- *Organizational hierarchy*: Not only the internal hierarchy should be represented but also the structure of the partner organizations. This allows to communicate with specific roles in specific departments, for example sending information to the clerk in the purchase department of an organization.
- *Persons*: Everybody, who participates in a business process has an entry into the person table. This gives the system a unique identifier and assigns some default rights or roles to the person. The following information is stored about persons.
  - *rights*: what may the agent do and see: his own worklist, the history of processes (full or condensed), the worklists of other persons of the same department, etc.
  - *address*: e-mail address, fax number, postal address, or HTTP URL may be possible entries in the address fields.
  - *client-type*: specifies in what form the tasks will be delivered: as links to the HTML client or as folder of forms.

The emerging directory of organizations and people will be useful for other purposes too and may be a component of the wfms or an integrated application.

4. Process definition

The definition of a workflow consist of three parts: the definition of the individual tasks, the definition of the sequence, and the types of data used in the process. In the following we use WDL (Workflow Description Language), the script language for describing workflows of the wfms Panta Rhei [2] for our examples. WDL defines the structure of the steps in the workflow using control structures for loops, branches, and parallelism. Every such step specifies *who* does *what* with *which data*, for example:

```
sek make_approval(application);
```

This step defines that the role **`sek`** performs the task **`make_approval`** using the form **`application`**. The agent of a task can be either a user, a role, an organizational unit, the agent of a previous task, or the value of a form field. This form of dynamic agent assignment allows the specification of ad-hoc tasks. The definition of tasks in WDL contains the specification of a post-condition, a maximum execution time, or a program, if the activity is executed automatically. The

form arguments can have the modes in, inout, and out, indicating whether the form is an input form, an input and output form (will be changed in the task) or an output form (gererated in the task).
The following example shows the definition of a simple order process:

```
process order_proc()
forms req recquote,
      order order_form,
      quote quote_form;
      invoic invoice;

begin
  all make_request(req) first;
  req.company request_quotes(req, quote);
  first:user check_quote(quote, order);
  order.company make_order(order);
  order.company make_invoice(invoic);
  sec pay(invoice);
end;
```

After the header with the process name the forms used in this process are defined as pairs specifying the form name followed by the form type.
In the first task the anybody (role **`all`**) can make a request and fill in the request form. The token first is a label used for referencing this step later. As agent of the next task a form-field is specified, the company field of the request form. If this field contains a reference to an external agent, the task is executed outside the organization. The task **`request_quotes`** must be defined with one input form, the request for the quote, and an output form, the quote. In the execution the process will send the input form to the company (the input to the remote task) and wait until the output form is received. In the next task the user of the first task has the possibility to check the received quote and fill in the order, which is delivered in the next step. The following task, make_invoice is again executed on the remote system: It has only an output form, so the process will wait until this form is received. The monitoring component will report this status of the process, allowing the business partner to see what is expected at this point. In the last step the **`sec`**[retary] pays the invoice.
Also part of the process specification is the interface definition, indicating what type of process is started when a form of a particular type is received. For example, forms of type order will start a selling process.

In the process definition we make no distinction between local and non-local agents or between the types of clients the agents use. However, the workflow engine interpreting the process definition must distinguish between the different agent types and act accordingly.

5. How the wfms interprets non-local agents

When the workflow engine starts a new task and assigns an agent to it, the following procedure is started:

if *agent is extern* **then**
    case *type of address and client is:*
        email, online: send an email alert with links to the HTML-client;
        HTTP-address: put the forms in a folder and make file upload;
        email, offline: put the forms in an email and send;
        fax: make image from forms content and send;
        letter: put content image in worklist of secretary;
    end case
end if

Briefly, there are two possibilities: send the forms or send an alert message containing a link to the HTML-interface. Accordingly, the task is finished in two ways:

- the worklist client finishes the task: this is the case for internal users and external users working with the HTML interface.
- The task is finished, when the wfms receives the forms of a task handled externally.

The reaction of a wfms on receiving of forms is shown in the following algorithm:

if *the forms belong to a running process* **then**
    if *the process is waiting for these forms* **then**
        continue process
    else exception1
end if
else if *interface defined for these forms* **then**
        start new process
    else exception2
    end if
end if

When the received forms belong to a running process the process is continued, otherwise a new process is initialized. When a form is received, for which no process is defined, or which is unknown, an exception is thrown. The normal handling of such an exception would be to start a new default process where some default recipients get these forms in the worklist.

The functionality of the wfms is the one side of the interaction, on the other side the client must also act accordingly:

1. The Web client is started following a link in a notification email. Because the user works interactive this client uses the same wfms API as other user interface clients.
2. The offline client must have the following functionality:
   - receive a form folder,
   - show the process and task information,
   - allow editing of the forms,
   - send the forms when the task is finished.

As transport mechanism for sending forms to the offline client, either email or HTTP would be possible.

6. Form mapping

The communication via forms anticipates that the organizations use standardized forms. When communicating with a customer via the Web client, proprietary forms could be used. Also when using the offline client – the form description can be sent together with the forms. But communication between different wfmss, where the process execution can be controlled by the form content, a standardization is necessary. A good choice would be the EDIFACT standard of the United Nations [6], which defines the structure of several dozens of messages for electronic commerce. Examples are request for quote (RECQUOTE), request for reservation (RESREQ), or invoice message (INVOIC).

However, EDIFACT documents have a severe drawback, they are well suited for ordering thousand windscreens but nobody wants to use them when ordering a book or a pizza. For such cases a form mapping procedure must be defined, mapping the fields of proprietary forms to the fields of EDIFACT forms and vice versa.

7. Conclusions

IT-support for performing business processes should not end at organizational boundaries. We proposed a system for telecooperation which realizes communication and cooperation with different kinds of external partners in a uniform way. Core of our system is the usage of a form-exchange mechanism as communication infrastructure. In these forms case date and process data are represented in a uniform way. We strongly support the use of HTML and XML forms to make the system more open since these forms are both readable by humans and IT-systems like workflow management systems. We described how the processes can be modeled and defined independent whether workflow participants are internal or external. The Web-based workflow client we developed enables arbitrary users to start workflows and to participate in workflows. The form interface recognizes forms sent to the workflow management system and stores the contents in the database, and notifies the wfms which then continues the appropriate process. On the output side the form interface transfers forms into the format expected by external partner and sends the form over the communication channel specified in the organizational database.

8. References

[1] T. Berners-Lee, R. Fielding, and H. Frystyk: *Hypertext Transfer Protocol - HTTP/1.0*. RFC 1945, http://www.w3.org/Protocols, 1995

[2] J. Eder, H. Groiss, and W. Liebhart: The Workflow Management System Panta Rhei, in *Advances in Workflow Management Systems and Interoperability*, Springer-Verlag, 1998

[3] H. Groiss, J. Eder: Interoperability with World Wide Workflows In: A. Ertas, C. V. Ramamoorthy, M. M. Tanik, I. I. Esat, F. Veniali, Taleb-Bendiab (eds.):1st *World Conference on Integrated Design & Process Technology*, Austin, TX, 1995

[4] T. W. Malone and K. Crowston. The Interdsciplinary Study of Coordination. *In ACM Computing Surveys*, 26(1), March 1994.

[5] Object Management Group: *Workflow Management Facility - Request for Proposals*, OMG Document cf/97-05-06, 1997

[6] UN/EDIFACT Rapporteurs: *United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport*, http://www.unece.org/trade/untdid/welcome.htm, 1997

[7] WfMC: *Interoperability - Abstract Specification* (WFMC-TC-1012, 20-Oct-96, 1.0) http://www.wfmc.org/, 1996