8-15-1997

End-User Workflow Management on the Internet

Sho-Huan Tung
National Yunlin Institute of Technology, tungsh@mis.yuntech.edu.tw

Jun Jun Wu
National Yunlin Institute of Technology

Shyh-Chau Wang
National Yunlin Institute of Technology

End-User Workflow Management on the Internet

Sho-Huan Tungand Jun Wu and Shyh-Chau Wang
Department of Management Information Systems
National Yunlin Institute of Technology
Touliu, Taiwan, R.O.C.
E-mail: tungsh@mis.yuntech.edu.tw

Background

Workflow management refers to the automated coordination, control, and communication of people, for the purpose of completing some business process, using a proactive computer system to manage the flow of work among participants (Hales and Lavery, 1991; Reinwald, 1994; Workflow Management Coalition, 1993). In order to support the development of workflow, a workflow management system usually provides an extensive set of tools to support the modeling, design, specification, implementation, and execution of workflow programs (Jablonski and Bussler, 1996). Most workflow management systems are designed for professional programmers who need to design and implement long-lasting, large scale, or complicated workflow applications. Action Workflow (Action Technologies, Inc., 1993), FlowMark (IBM Corp., 1995), and InConcert (McCarthy and Sarin, 1993) are just a few examples of a huge list of commercial workflow products.

Many types of workflow, however, may not be large or long-lasting. They may be simple, short-termed, or even casual. For example, faculty members in a department may have the need to find a mutually convenient time to have lunch together at some appropriate restaurant. A professor may want to collect skills and experiences from graduate students in order to schedule interviews with possible candidates for research assistant positions. Traditional workflow management systems are ill-suited for these kinds of occasional workflow programs.

This paper introduces the EUFlow end-user workflow management system that allows end-users with elementary programming skill to write workflow programs. EUFlow workflow relies on e-mails to coordinate participants' activities by sending and receiving information needed for a workflow. EUFlow consists of the EUWoM end-user workflow model, the EULAN workflow composition language, the EUENV visual programming environment, and the EURUN run-time system. The remainder of this paper briefly describes the EUFlow system and its current status.

EUWoM End-User Workflow Model

Message rounds are basic building blocks of an EUWoM workflow. A message round contains a software process called a *message handler*. The message handler is responsible for sending messages to one or more receivers. A message can be a *notification*, which does not require a reply, or a *message form*, which contains various *fields* for the receiver to complete. For simplicity, a message round can define only one message form. Different copies of this same message form, however, can be sent to more than one receiver. Upon receiving a reply, the message handler process's information contained in the form. When a condition for terminating the message round is met, *flow control rules* are used to decide how to continue the workflow. A message handler is usually connected to a database to obtain e-mail addresses of its receivers or to assist information processing. Figure 1 presents basic components of a message round.

Workflow control rules can specify the conditions of when to stop message handling, to start other message rounds, or to store processed data to the database. A message handler may be stopped at some specified time that matches the deadline of a work. It may run continuously until receiving replies that satisfy certain conditions. A group of message rounds may be *joined*. After joining, other message rounds may be *forked* to continue the workflow. Message rounds may run concurrently or sequentially.

EULAN Workflow Composition Language

The definition of a workflow can be specified by the EULAN workflow composition language. EULAN consists of two levels, the composite level and the message round level. The composite level specifies interfaces among message rounds to allow individual message rounds to be combined to form a workflow. The message round level allow programmers to write message rounds adhering to the EUWoM workflow model. After translation, an EULAN program can be executed on the EURUN run-time system to carry out a workflow.

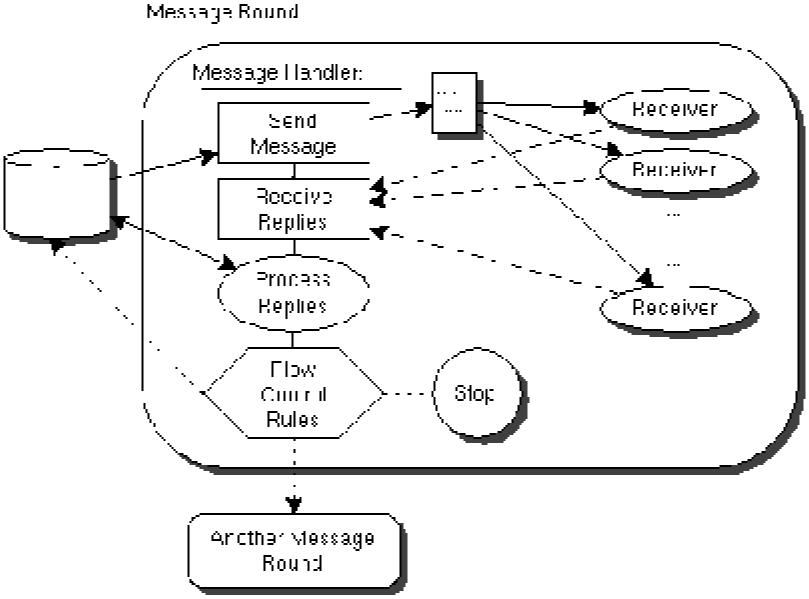


Figure 1: A Message Round

A common task performed by message rounds is to get information from a database, to collect information from message receivers, and to store collected information back into the database. Databases, tables, fields, and their attributes are therefore important information specified by message round interfaces. Two message rounds can be connected if one can provide sufficient input source to trigger the execution of the other.

EUENV Visual Programming Environment

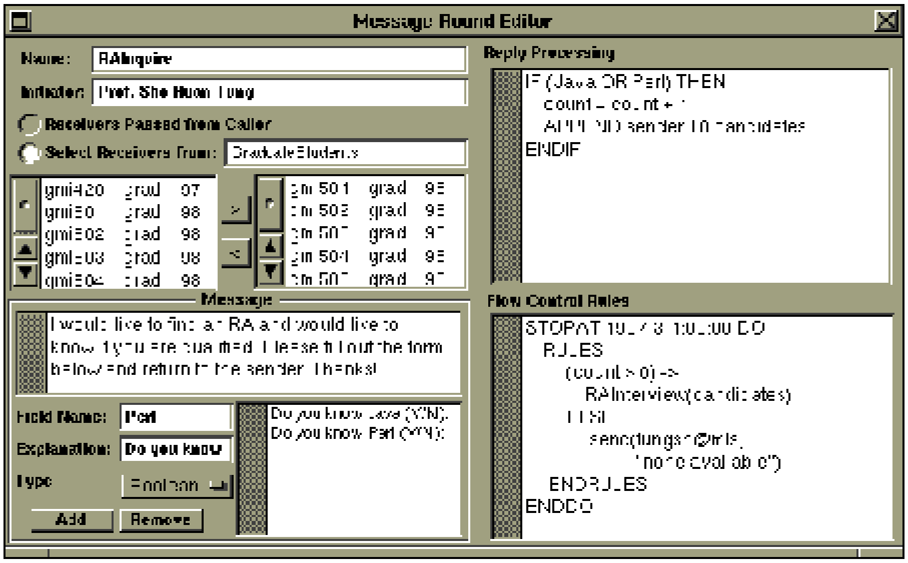
The EUENV visual programming environment allows end-users as well as professional programmers to write EULAN programs using convenient user interface tools. EUENV also has two levels. End-users can use the composite level to combine existing message round components to construct a workflow. Professional programmers can use the message round level to specify computations actually carried out by a message round. Figure 2 presents an EUENV specification of a message round in a workflow program that assists a professor to find a qualified research assistant. A qualified candidate for the research assistant position should be a first year graduate student and has programmed Java or Perl.

The **RAInquire** message round sends a message to all the first year graduate students selected from the **GraduateStudent** database. The message contains text and an information request form. After receiving the message, the receiver fills out the form and replies to **RAInquire**. The **Reply Processing** code specifies how to handle the message. **RAInquire** calls **RAInterview** and passes it the email addresses of qualified candidates before stopping itself at March 1, 1997. **RAInterview** can send out further instructions for scheduling interviews.

EURUN Run-Time System

The EURUN run-time system is essentially a special purpose mail server which understands how to process message replies sent from receivers. A replied message contains the name of the workflow, the name of the current message round, and form information provided by the receiver. Upon receiving a message, EURUN execute the message round responsible for handling the message and passes it information contained in the mail.

One way to implement EURUN is to store information about workflow programs and message rounds in a database. The EURUN mail server uses this database to obtain detailed information about workflow programs, messages rounds, their interfaces, and execution sequences. The composite level of EUENV programming environment can provide an interface to allow end-users to create a workflow program by filling in information in associated database tables using existing message round components.



Name: RAInquire

Initiator: Prof. She Huan Tung

○ Receivers Passed from Caller

● Select Receivers From: Graduate Students

gmi420	grad	97
gmi50	grad	98
gmi502	grad	98
gmi503	grad	98
gmi504	grad	98

gm501	grad	95
gm502	grad	95
gm507	grad	97
gm504	grad	95
gm507	grad	97

I would like to find an RA and would like to know if you are qualified. Please fill out the form below and return to the sender. Thanks!

Field Name: Perl

Explanation: Do you know

Type: Boolean

Do you know Java (Y/N):
Do you know Perl (Y/N):

```
IF (Java OR Perl) THEN
   count = count + 1
   APPEND sender TO candidates
ENDIF
```

```
STOPAT 19[illegible]/8 1:00:00 DO
   RULES
      (count > 0) ->
         RAInterview(candidates)
      ELSE
         send(tungsh@mis,
               "none available")
    ENDRULES
ENDDO
```

Figure 2 EUENV Message Round Level

Current Status and Future Work

At present time, we have finished the design of EUWoM and the message round level of EUENV. Detailed design of EULAN and the composite level of EUENV is underway. The most difficult aspect in designing EULAN is the trade off between simplicity and power. Fortunately, we can always implement a simple end-user programmable subset of EULAN in EUENV while providing the full power of EULAN to professional programmers for more complicated workflow applications. In addition to designing the message round level of EUENV, we are also studying the possibility of using existing programming languages to write message rounds and allowing them to be combined by the composite level of ENLAN.

We are implementing EUFlow in Perl and Java. Perl and Java are commonly used in world-wide-web applications. Perl is primarily used for CGI programming and Java allows programs to be execute in browser. EURUN will be implemented in Perl. Message rounds of a translated EULAN program can be stored in a database and executed by the EURUN run-time system. Java is used to implement the EUENV programming environment as well as tools for monitoring and administrating work lists. A working prototype of EUFlow running on Unix systems is expected at August 1997.

Besides simple or occasional workflow programs, we believe EUFlow can also be used to construct a wide range of applications. A virtual store can be implemented in EUFlow by sending email order forms to registered customers and processing the replies automatically. An instructor can also write an EUFlow program which sends and grades assignments automatically.

References

Action Workflow, *White Paper*, Action Technologies Inc., Alameda, CA, USA, 1993.

IBM FlowMark. Workflow Modeling. Release 1.1., IBM Corporation, Viena, Austria, 1995.

Hales, K. and Lavery, M. *Workflow Management Software: the Business Opportunity*. Ovum Ltd., London, UK, 1991.

McCarthy, D. and Sarin, S. Workflow and Transactions in InConcert, *Bulletin of the Technical committee on Data Engineering*, Vol. 16, No. 2, 1993.

Reinwald, B. "Workflow Management," *Tutorial 13th IFIP World Congress*, Hamburg, Germany, 1994.

Workflow Management Coalition Application Programming Interface (WAPI) Specification, WFMC-TC-1009. Workflow Management Coalition, 1995.

Jablonski, S. and Bussler, C., *Workflow Management - Modeling Concepts, Architecture and Implementation*, International Thomson Computer Press, 1996.