Management Approval Routing System – MARS

Project Proposal

By

Arun Shastry

Ross Scott
Introduction

MARS is a web enabled workflow automation tool that facilitates requisition processing in a typical enterprise. The system will use an Oracle 8i database and will be deployed as a web application on an Allaire JRun server running on a Windows 2000 machine.

The Need for MARS

Any medium to large enterprise will need some sort of an automated tool to help employees request things – PC’s, software, furniture, job openings, and other such items. Also, these enterprises will have a relatively complex organization structure and each employee in the management channel has an approving capacity depending on his/her rank. Depending on the amount, the requisitions may need to be approved systematically by each of the managers in the org structure before becoming an order or procured.

In such an environment, it gets extremely difficult, even impossible, to manage requisitions manually. Also, accountability would become an issue if we cannot track the lifecycle of such requisitions. MARS will address these issues, and others, by automating the entire approval process and could be extended to a sophisticated B2B system so that an approved requisition can be converted to an order and sent to the corresponding vendor.

Specifications

MARS is a J2EE application integrated with an Oracle 8i database and will be developed and deployed on an Allaire JRun java server. Below is a high level schematic of the development environment:

![Diagram of MARS development environment]

Note: The above is the development system that will also be used for presentation. However, in a typical production environment, this system will be tiered and the clients/browsers will be on separate machines as will the Oracle 8i server.

At the core of this system is the application server and our middleware developed as servlets and web pages. The JRun server is configured with appropriate JDBC connectivity to the Oracle server. The application logic will consist of both servlets and
database objects like stored procedures, views, sequence generators, etc. The data model will consist of the following logical entities:

- Employees identified by userid’s
- Requisitions identified by req_id’s
- Purchasable items identified by part_id’s
- Supporting reference data (e.g. approval limits, actions, etc.)

**Features/Functionality**

Due to the limited time to develop this system, below is a list of features and functionality we intend to implement as part of this project:

1. Only tangible assets supported in the first version (e.g. PC’s, furniture, etc.)
2. Create/Edit requisitions
3. Track requisitions submitted and in case of managers, track all requisitions submitted by his/her subordinates
4. View all historical requisitions
5. For managers, approve or reject requisitions awaiting action. Approving a requisition will move it ahead in the management approval channel, until it reaches the last person in the list at which time it becomes fully approved; rejecting it will send it back in the queue and the manager has the option of selecting the person in the list to bump it back to. In either of these scenarios, the manager can input comments.
6. Requisitions can contain multiple items
7. The following organization structure will be used for demo purposes; this can be changed at any time:

   Employee --> Division Manager --> Director --> VP

8. Requisitions up to $500 need to be approved only by the employee’s supervisor.
9. Requisitions between $501 and $5000 can be approved by the division manager, requisitions between $5001 and $10,000 require a director’s approval, and all requisitions over $10,000 require a VP’s approval. These limits are for our demo purposes only and can be changed any time.

**Process Flow Diagram**
1. User Inputs Requisition

2. The actual authentication takes place
   
   a. Demote Request
   
   b. Request Denied

   c. Request Granted

   i. 1. Set ‘STATUS’ to “Waiting for Authentication”
   
   ii. 2. Send email to Buyer and Requestor

   3. Buyer sends requisition to vendor and fills out additional information

   4. 1. Set ‘STATUS’ to “Waiting Delivery”
   
   2. Send email to Buyer and Requestor

   5. Delivery comes in

   6. Buyer closes out requisition
User Interaction phases:

1) Requestor submits requisition request:
   a) Information is collected: (item drop-down lists used):
      - Vendor (required if quote entered)
      - Vendor Contact (required if quote entered)
      - Requester’s Phone Number (required)
      - Item to be ordered (required)
      - How many we are ordering (required)
      - Price (required)
      - Comments
      - Requester (required)
      - Date/Time (required)
      - Justification (required)
      - Plant (required)
      - Receiver (required)
      - Delivery info
        - Plant (required)
        - Receiver (required)

2) Buyer submits actual request to Vendor
   a) Actual Information about requisition
      - Date Buyer sent Requisition to Vendor (required)
      - Estimated retrieval date (required)
      - Requisition # (required)
      - Which acct payment is divided into (required)

3) When part comes in
   a) Information Collected
      - Purchase Order # (required)
      - Total order cost (X Price) (required)
      - Voucher Date (required)
      - Expense Account (required)
      - Closing Comments (required)
      - Receival date (required)
      - Date Paid (required)
Tentative Project Plan and Milestones:

**10/9/2001**
Project Proposal

**10/16/2001**
Preliminary datamodel ready
Become proficient with Jrun and Oracle

**10/23/2001**
Finalized datamodel
Development environments ready
Datamodel implemented on development environments
Servlets completed:
- Authentication

**10/30/2001**
Test data ready
Any changes to the datamodel
List database objects required – stored procedures, triggers, views, sequence generators
Servlets completed
- Buyer sends request to Vendor

**11/6/2001**
Implement all database objects and interface with the application
Servlets completed
- Buyer receives product(s) and closes requisition

**11/13/2001**
Servlets completed
- Initial User interface

**11/20/2001**
Servlets completed
- Authentication

**11/27/2001**
Alpha Testing

**12/4/2001**
Finalize Documentation

**12/11/2001**
Prepare for and give demo