



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Framework Model for Workflow Management System

H.H. Owaied, H.A. Al-Kareem Farhan and Y.W. Hudeib
Faculty of Information Technology, Middle East University, P.O. Box 42, Amman 11610, Jordan

Abstract: This study presents the proposed a framework model for design and implementation of workflow. The model consists of three modules these are; dynamic process module, database module and activity transaction module. The proposed model works through the interaction with users using the web application interface in order to enable users using the facilities provided by the web application such as the user can login to the system either through internet, intranet or extranet. The Database module initiates a connection between all modules. The proposed framework model can made the dynamic workflow engine manages the entire life-cycle of electronic forms and the processes involved. The electronic form processing enables organizations to easily collect information through online mode and process the requests in accordance with predefined business procedures. The proposed model can facilitate the communications between the groups within an organization in efficient manner of movement of information dynamically through the use of dynamic process module in order to run their businesses effectively.

Key words: Business management, e-commerce, information technology, workflow management system

INTRODUCTION

In this century the application of infrastructure of Information Technology become very importance and most of the enterprises, organizations willing to improve their businesses through the new philosophy of business reengineering. The processing of information in any organizations required some tools to cope with the reengineering their businesses such as using automation of their existing processes and react quickly to the changes in the business environment. Usually the communications between the groups within an organization depend on the efficient movement of information in order to run their businesses effectively. The automation of business processes, in whole or parts, where documents, information or tasks are passed from one participant to another to be processed, according to a set of procedural rules is called Workflow (AL-Rossais, 2007). There are many definitions of Workflow such as is the numbers of process-modeling techniques which are available to define the detailed routing and processing requirements of a typical workflow. Also, there is another description for workflow; it is the automation of business processes, in whole or part, during which documents, information or tasks are passed from one participant to another for actions, according to a set of predefined procedural rules. Workflow can be described simply as the movement of documents and tasks through a Business process.

On other words, it can be a sequential progression of work activities or a complex set of processes each taking place concurrently, eventually impacting each other according to a set of rules, routes and roles (Repetto *et al.*, 2003). Because business information moves quickly and the manual collection of data can lead to reduced productivity and costly delays. The proposed solution is the management of the entire lifecycle of forms and the processes involved which is called Workflow Management System (WFMS) (Momotko, 2005). Therefore the Workflow Management System can be defined as a system allows organizations to define and control the various Activities associated with a business process. In addition, many management systems also allow a business the opportunity to measure and analyze the execution of the process so that continuous improvement can be made. In order to improve the capability to respond dynamically to process changes, dynamic workflow model base on flexible activities and historical information have been introduced. In early 90s increasing number of workflow applications specified two new important requirements for WFMS, namely integration with existing applications/services and co-operation among different WFMS (Li and Du, 2009). The first Application integration is a major trend in information technology today. Three of the major trends in information technology today are the Web, enterprise software packages and application integration. The Web provides an environment that can link a company's

customers, suppliers, partners and its internal users. Enterprise software packages offer an integrated environment to support business processes across the functional divisions in organizations. Some packages, like Enterprise Resource Planning (ERP), for example SAP R/3 and BaanERP, manage back-office requirements, while other packages provide front-office capabilities, e.g., customer services (Teach Teams.com). Common to Web applications as well as enterprise software packages is the need for application integration.

Application integration is required to connect front office systems with back office systems, to transfer business processes to the Web and to create extended supply chains involving customers, partners and suppliers (Paul and Erik, 2000).

Application integration is also needed for wrapping legacy systems and for migrating to new environments. The demand for application integration is also fuelled by the move to process orientation in many organizations, such as creating distributed data-sharing applications. It is an integral part of the .NET Framework, providing access to relational, XML and application of DBMS (msdn.microsoft.com, 2010). Traditionally, organizations have been functionally divided, i.e., companies have been separated into departments such as market, production and service. However, the functional organization has been shown to have a number of weaknesses. In particular, it requires a huge administration to handle issues crossing functional borders and considerable resources are allocated to tasks that do not create value. Using the cooperation mechanism between different WFMS, the problems of a functional organization may be solved. This mechanism defined as process participates in inter-organizational processes among different WFMSs and specifies the associated arguments in and out. The mechanism, also allowing dynamic changes and reconfiguration, the cooperative process across organizations becomes simple, faster and flexible (Jian *et al.*, 2007).

THE PROBLEMS OF WFMS

There are many problems related with the WFMS, these are:

The workflow adaptability and standardizations: The efficiently movement of information within an organization usually depend on the adaptability of workflow and standardizations used to run their businesses effectively. Inefficient movement of information and manual collection of data lead to reduction in productivity and costly delays (Adams *et al.*, 2003).

Static form and application: This is restricted for the end user, Because when need to make any change for this

form, must refer to the programmer to insert or drop any component, so to solve this problem propose dynamic form and application and allow to the end user to use the form and application by more flexible, this is the first step apply the dynamic workflow engine (Charoy *et al.*, 2006; Salle *et al.*, 2006).

Static activity: The activities are deployed by programmer and are used for specific aims, which means can't use that process for general.

Static rules: The rules are business logic issues that made by programmer using certain programming language such as XML language. The rule can develop the activity, but it can program for once. The rule is reflecting to the activity from the engine side, in this thesis, we will discuss how the user can make dynamic activity then reflect to become dynamic rule.

In order to improve the capability to respond dynamically to process changes, dynamic workflow model base on flexible activities and historical information have been introduced. The proposed model is based on human behavior during accomplish a certain task. The human behavior consists of the following steps:

- **Step 1:** The normal user choose the type of process that he need (the user in this level called initiator), then he fills the needed data that concern the process type and form, in this level it is a normal round, normal users can't change in this form such as add any component, on other word they can't use the dynamic form that will explain later and it is the first step to make a dynamic engine, the dynamic engine is a procedure concern with permission and organize the authorization then it will send all data to the manager who have the more permission
- **Step 2:** The Manager will open the form that opened in the above point by the initiator, the manager have a dynamic form that assist him to check about finished works, if it is not ok he ask from the initiator to make some changes and the status in this time will be (No), that is not accept from the system, the initiator make changes that asked and his role will end, in this level the manager can add any component, text and attachment that will serve other user to understand the process
- **Step 3:** After the process sent and it receive from the first user, he browsing the content and he has (Yes, No) options, after he decide the appropriate decision he send it to complete the path of process. When the real path ending the system will check about this process if there is anyone was select No the process

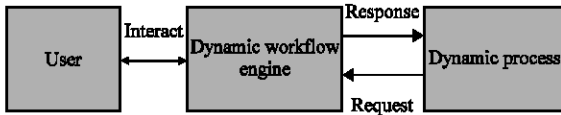


Fig. 1: Context flow diagram

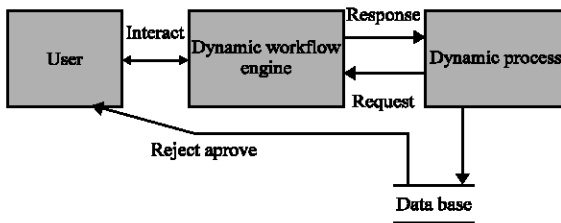


Fig. 2: Level (0) logical flow diagram

will return to the initiator, he will make the suit changes either in content or in the path by adding or deleting users from the roles

Figure 1 presents the context flow diagram for the proposed model and Fig. 2 presents the o level of logical flow diagram. The proposed model will be described in details in the next sections.

THE ARCHITECTURE OF PROPOSED MODEL

The proposed model consists of three modules as seen in Fig. 3, these are:

- Dynamic process module
- Database module
- Activity transaction module

The Dynamic Process Module consists of three Parts these are:

Web application interface: The proposed model of the system works through the interaction with user. The interface includes web application as well contains login screen, after the user input his own user name and password. Then it will verify to take the own authentication from database depend on permission associated for each user that given before. User can select the type of process as (vacation, signature approval. etc).

As shown in Fig. 4. The user will be able to use all the facilities provided by the web application. The web application allows the user to login to the system by (internet, intranet or extranet).

Generating dynamic process form: Figure 5 shows the process of generating dynamic process form through the user fills the specific field and if he want to add any data that doesn't have location as (Text, Label and any component) he presses on the generate text bottom from XML page by specific code, using the algorithm Generate Text Fields.

The following is a Dynamic Generate Text Fields algorithm which gets a sequence of names.

```

Start,
Initialization,
Field_Name = "txt"+I;
Add (Field_Name)
Conert Form Fields to XML Schema
Create Element Node:
Parent_Node.
<TaskInfo TaskID="Value">
Child_Node.
< Remaining_Fields > Values </ Remaining_Fields>
Save Task XML Schema in Tasks Table
If (Tasks_Insert (TaskID, TaskXML, UserID)>0)
lblMsg.Text = "Add New Task Successfully
UserID: Login User ID
End.
  
```

Creation dynamic activities: This part concerns for assigning the activities and their path to the users in order to generate the dynamic activities as seen in Fig. 6. Those users are the peoples involve with every activity, but if any person does not exist in the database the module will added by the following algorithm.

```

If check_Exist (login Name) =true Then
Msg = "This is user Already Exit"
Else
Insert_user (UserID, EmpName, LoginName, Password)
End IF
  
```

The authorized person (Manager or Administrator) retrieve the static activates from database then according to the requirements produced through the user interface then the dynamic activates will be generated. Through the dynamic activity creation processes, authorized person can assign the paths and users which are involve in any specific process and assign the flow sequence together with the date and time as shown in Fig. 7 using the following algorithm.

```

For I as integer = 0 to lbUserRole.item.count-1
User ID: lbUserRole.items (I).value
Insert_taskRoles (RoleID, TaskID, lbusersRole.item (I).value, RoleOrder, Status)
Status: = Initially = Null
Insert_ShowTask (ID, TaskID, lbuserRole.item (i).value, IsShow)
IsRead: Initially = False
  
```

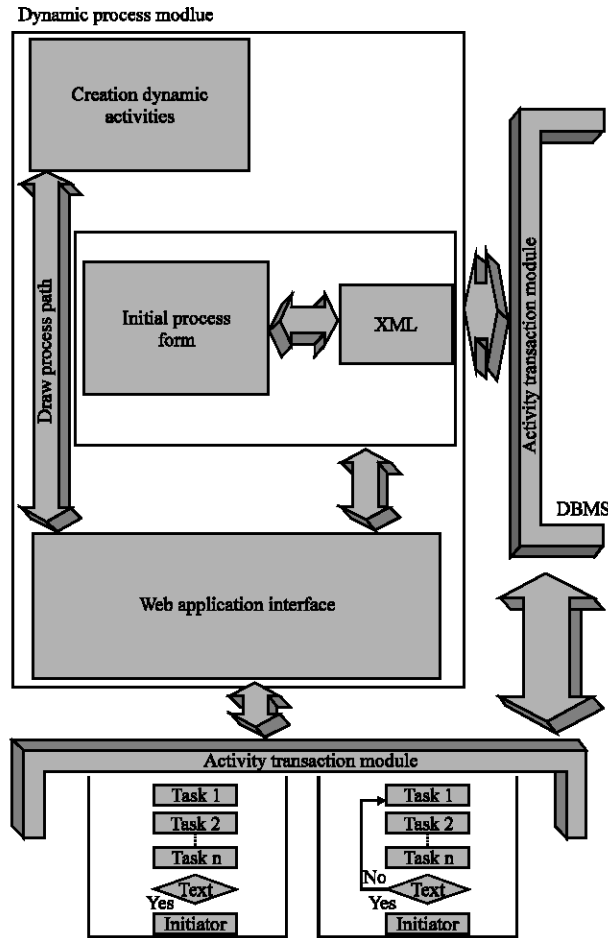


Fig. 3: Architecture of the proposed model

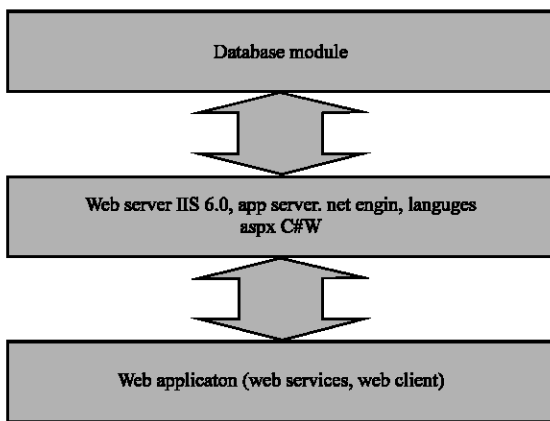


Fig. 4: Web application interface

The explanation of the above algorithm:

- There is an index for every user depends on the place in the list

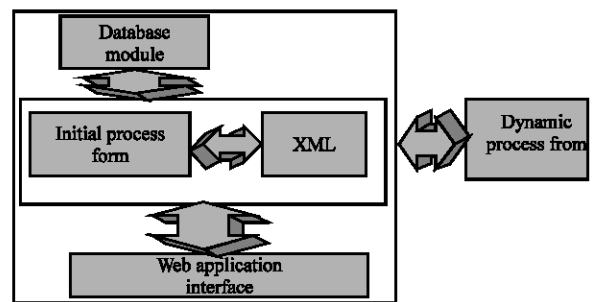


Fig. 5: Generating dynamic process form

- `lbUserRole.item.count`: take the value of the first index in the list
- `UserID: lbUserRole.items (I).value`: user value depends on the place in list
- `Insert_taskRoles (RoleID, TaskID, lbusersRole.item (I). value, RoleOrder, Status`: store in database server where it stores the RoleID, TaskID and Path that the process will move on it

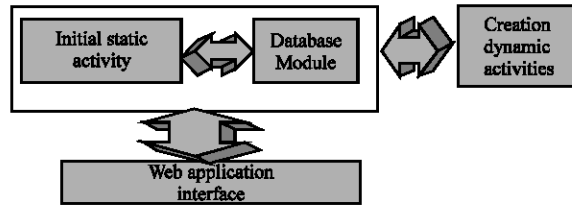


Fig. 6: Creation dynamic activities

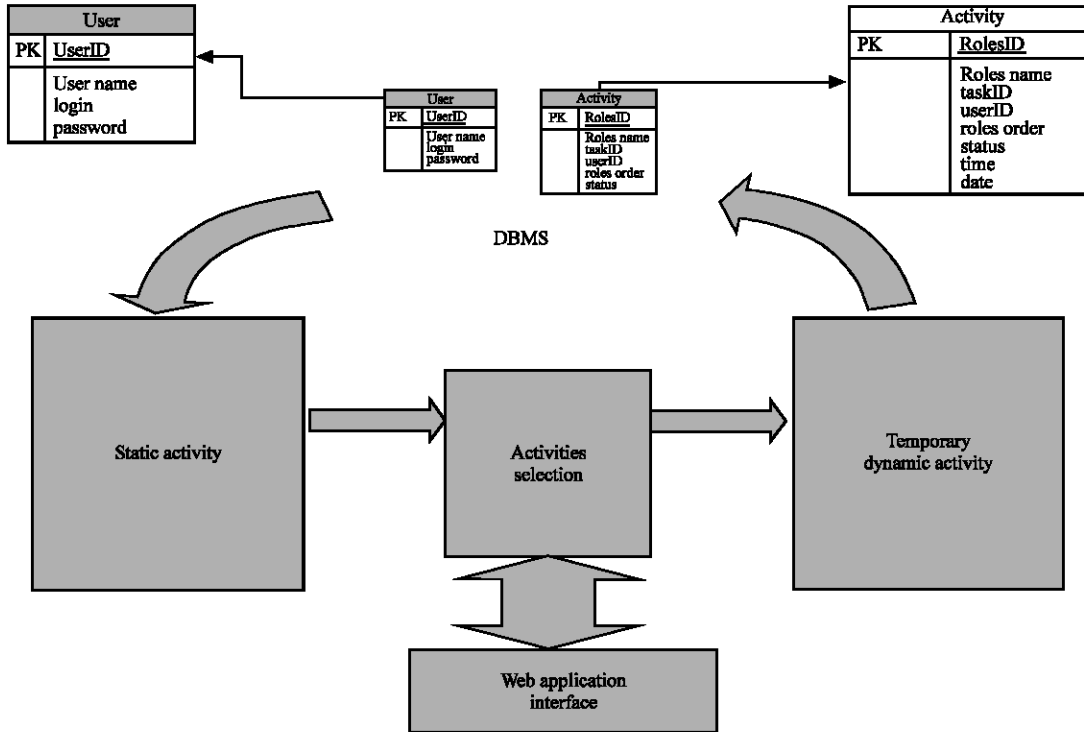


Fig. 7: Creation dynamic activities

- IusersRole.item(I).value: store the type of process and the value of user depends to the list and Role Order, also it store the User ID and task status that means does the user open the task or not

Database module: Using the Data Access Responses in order to initiates a connection between all Modules and Database by using ADO.Net as shown in Fig. 8.

Figure 9 presents the ER diagram for the tables used in the proposed system which consists of the following tables.

- Dynamic activity table: It is the table that store the value of path that drawn from the first user and it contain the role name and number and it is store the sequence if users that the process will move to them

- Process form table: It is store the task ID and name and the name f initiator and all data are saved as XML
- Show task table: It is store the values that mention to the status are open by user or not
- Static activity table: it is contains the users that involve in specific process

Activity transaction module: After the process build by authorized person, the process starts by move to first user he decide about the status by approve or reject and so on till the end of process path. If all users are agreed it will be finished and return to initiator by approve status as shown in part 1 of Fig. 10. But if the process has any reject status it will be returned to initiation to change and modify as request then resend it again to all users as shown in part 2 of Fig. 10, it can presented as If Clause as follows:

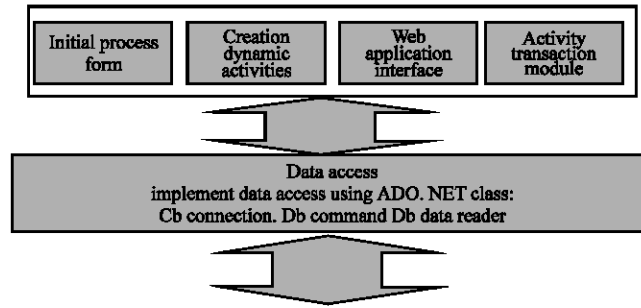


Fig. 8: Database module

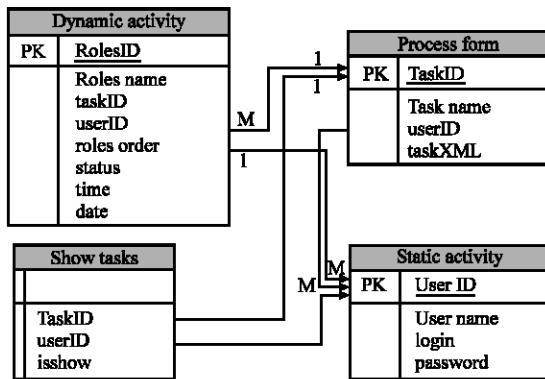


Fig. 9: ER diagram

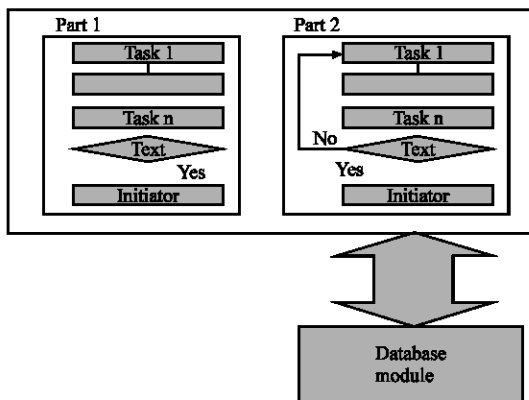


Fig. 10: Activity transaction module

```

IF (Task1^Task2^... ^Taskn = "True") Then
Return Initiator "Process successful Approval"
Else (Task1^Task2^... ^Taskn = "False") Then
Return Initiator "Process Round 2"
And the below algorithm clear the functionality:
User Receive Task
If Receive _Task (userid) = "get records" Then
{ Receive new task if status of current userid roleorder-1=True or False
If show_task (taskid, userid) = True Then
{ Update _show task (taskID, userID, True)
Update show task isshow=True Where taskID and userID
If Update_taskrole (taskid, userid, True) Then
{ GO to Next user order in this task
Else Go to next user order in this task and to user

```

```

Roleorder=0} End if Return Receive _Task (Next User)
Else Return show task (Waiting ..... ) End if} End if}}

```

CONCLUSION

In this study presents many problems, such as workflow adaptability, Static activity and Standardizations. The proposed model can solve those problems through the application of dynamic process module, generating dynamic process form and creation dynamic activities

The proposed model of workflow management system is allowed organizations to define and control the various Activities associated with their business processes. In addition, facilitate many management systems the opportunity to measure and analyze the execution of the process so that continuous improvement can be made.

REFERENCES

- Adams, M., D. Edmond and A.H.M. ter Hofstede, 2003. The application of activity theory to dynamic workflow adaptation issues. Proceedings of the 7th Pacific Asia Conference on Information Systems, July 10-13, Adelaide, South Australia, pp: 1836-1852.
- Al-Rossais, N.A., 2007. Developing an Enterprise Workflow Solution. King Saud University College of Computer and Information Sciences Department of Computer Science, Saudi Arabia.
- Charoy, F., A. Guabtani and M.V. Faura, 2006. A dynamic workflow management system for coordination of cooperative activities. Lecture Notes Comput. Sci., 4103: 205-216.
- Jian, C.A.O., H. Zhao, J. Wang, S. Zhang and M. Li, 2007. Verifying dynamic workflow change based on executable path. Int. J. Intel. Control Syst., 12: 37-44.
- Li, P. and Y. Du, 2009. Modeling and design for dynamic workflows based on flexible activities. Inform. Technol. J., 8: 750-756.

- Momotko, M., 2005. Tools for monitoring workflow processes to support dynamic workflow changes. Scientific Council of the Institute of Computer Science and Polish Academy of Sciences.
- Paul and Erik, 2000. Design Principles for Application Integration 2000. Department of Information and Systems Sciences, Stockholm University/Royal Institute of Technology, Sweden.
- Repetto, M. and M. Paolucci and A. Boccalatte, 2003. A design tool to develop agent-based workflow management systems. Proceedings of the 4th AI*IA/TABOO Joint Workshop From Objects to Agents: Intelligent Systems and Pervasive Computing, Sept. 10-11, Villasimius, CA, Italy, pp: 100-107.
- Salle, P., F. Duvert, D. Herin and S.A. Cerri, 2006. Dynamic workflow in grid-MAS integration context. Proceedings of the 9th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Sept. 26-29, Timisoara, pp: 242-245.