

A Web-Based Rapid Prototyping Workflow Management Information System for Computer Repair and Maintenance

Ali H. El-Mousa, Zahra J. Muhsin and Majid A. Al-Tae
Department of Computer Engineering, University of Jordan, Amman 11942, Jordan

Abstract: Problem statement: Response to paper-based requests for computer and peripheral repair and maintenance has become very troublesome and slow due to the large demand and expansion at the University of Jordan. The objectives of this study were to: (i) investigate the current system processes associated with the paper-based workflow system. (ii) design and implement, using a suitable workflow management approach, a totally electronic alternative to improve performance. **Approach:** The methodology followed in the transform of the business processes from the paper to the electronic-based was using the rapid prototyping workflow management approach. This approach is seen to be especially effective in such cases where there is direct continuous interaction and involvement between the different stakeholders during the lifecycle of the project. **Results:** The system had been implemented and tested with the result that efficiency, accountability and response time have greatly improved in handling repair and maintenance orders. The transform from paper to electronic resulted in greatly enhanced user satisfaction especially since the developed system provided automatic feedback regarding order status. **Conclusion:** The design process and results provide a working blueprint for easy and quick various similar university-based business processes to be transformed to electronic. This should highly motivate internal in-house restructuring of business process to utilize technology and IT to enhance performance and accountability.

Key words: Electronic transactions, equipment management, communications technologies, information systems

INTRODUCTION

In today's information technology society, it is no longer acceptable nor is it cost conscious to keep to the old paper-based approaches in managing business processes. Therefore, it is not surprising that from early on, serious efforts were directed at utilizing whatever technologies were available at the time, to automate repetitive tasks using IT^[1].

In the field of equipment management, many efforts were attempted. A multi-user computer system for the management of hospital equipment including repair requests, inventory control, repair costs and service records was described by Whelpton and Cooke^[2]. The system uses a relational database together with personal computers. However, data entry transferred from study requests is done by only one user and there is no methodology to follow up requests or monitor activities. A similar system that manages information used for the management of preventive maintenance and equipment repairs at the University of California Berkeley was also described by Mudie and Chang^[3]. The system provides technicians and

management with a tool for monitoring equipment maintenance events, reviewing equipment history and resolving equipment malfunctions. The system allows equipment users to report faults which are automatically forwarded by email to maintenance staff. A relational database is used to store information regarding events. The system lacks online direct monitoring of events and status. It delegates the management of all events to the maintenance staff which reduces accountability.

Recently, it has been shown that projects developed using traditional software lifecycle models such as the waterfall model typically fail for several reasons, namely: output-driven orientation of design process, application backlog due to the document maintenance workload and user dissatisfaction^[4,5]. For small and medium sized systems, a development methodology based on rapid prototyping approach is intended to correct these shortcomings and to bring a software project closer to its user(s). After several iterations of the prototype, the developers have a better understanding of the requirements and the users have a better idea about how the system will eventually look

and feel. For large systems, the prototype based methods have difficulty in identifying the global aspects and boundaries of the project and design inconsistencies and shortcomings^[5,6]. However, despite their difficulties in large system developments, prototype-based methods have been a hot topic in software engineering for more than a decade. Different design models for workflow management systems have been reported in literature, using: Petri Nets^[7], agent technology^[8], knowledge-based^[9], document-based^[10] and more recently object-oriented modelling. When compared to other modelling methods, the object-oriented modelling approach offers improved flexibility and adaptability^[11].

In the present study, the architecture, development methodology, design models and implementation of a new web-based workflow management information system for computer repair and maintenance in the Faculty of Engineering and Technology (FET) at the University of Jordan is presented. The choice to transform the paper-based repair system business processes to an electronic form was first evaluated for suitability, based on the following factors:

- The processes required people to be notified of events as they occurred (request progress)
- Processes were approval-oriented (there is a management and control hierarchy in FET)
- The processes cross functional and/or department lines (repair requests are directed from departmental or managerial units to an independent repair and follow-up section)
- Processes where workflow will add value to stakeholders (expected reduction in time, effort and supervision and workload distribution)
- Automation of processes contributes towards university goals (there is a clear direction towards process automation using modern IT structures in the University of Jordan)
- Automation has potential for significant improvements (requests can be remotely monitored and repair workers are directly accountable. The system has the potential to grow and interlink with university IT systems)
- Processes are feasible to redesign (since the processes involved already are form-based with a clear flow, then it is totally feasible)

Problem background: The FET at the University of Jordan is one of the largest faculties in the university. It consists of eight academic departments and four associated administrative and technical units, of which

the centralized computer labs are the subject of this project. The working environment in all these departments/units is completely intra and inter networked. FET is planning to gradually transform all administrative and communicative work to become electronic. Part of the plan focuses on semi automating the workflow processes of maintenance and repair requests in the centralized computer lab to become completely web-based. This goal can be achieved by using an appropriate information system that is capable of dealing with different types of web-based transactions such as new user registration, submission of repair and maintenance orders, distribution of work items (tasks), viewing and reporting various reports. Such a system will enable all stakeholders to closely follow-up the progress of orders and get timely notification of status. Such a system will also allow monitoring of personnel performance and may help in optimizing the allocated jobs and time distribution.

The existing system involves a conventional paper-based process for order management. This process deals with order submission and workflow control activities as shown in Fig. 1. The main drawbacks for the existing workflow management system can be summarized as follows:

- Long lag time between order submission and service operation
- Lack of a balanced service staff load distribution facility, which leads to lax accountability and low productivity
- Nonexistent reporting of the different system variables which hinders the long term planning and expansion of the system
- Reduced reliability and productivity as there have been no procedures to follow-up the order and identify its current status
- Low user satisfaction due to all of the above

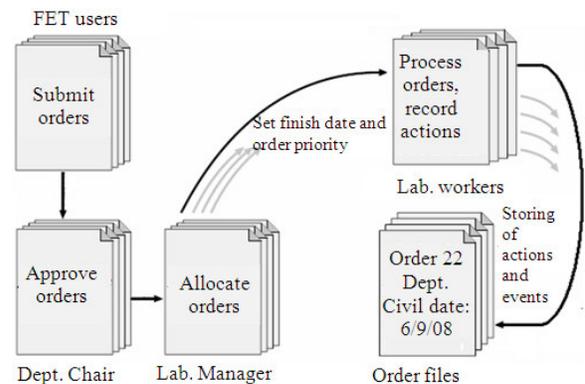


Fig. 1: The existing paper-based order process

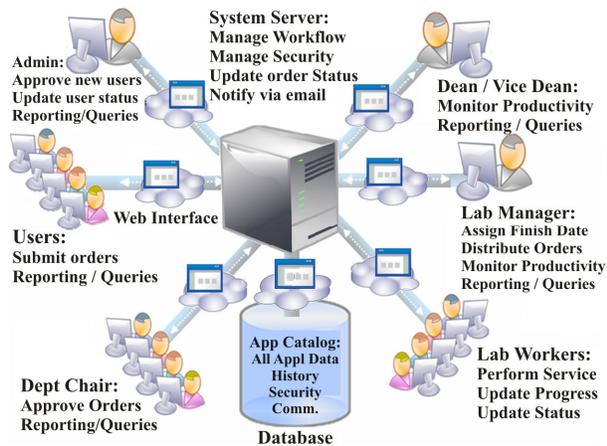


Fig. 2: Overview of proposed web-based WKFMIS

Proposed system overview: In addition to re-engineering the order management process mentioned above, the proposed system is supported by several newly added processes, namely: User management, report generation, real-time end-user follow-up and a configurable management path. Figure 2 overviews the proposed web-based workflow management system.

System architecture and features: Database Management Systems (DBMSs) are commonly configured using a three-tiered architecture which is appropriate for Internet applications. This architecture is therefore considered appropriate for the proposed workflow management system^[12]. The tiers of the proposed architecture can be seen as having clients at the front end, services at the middle tier and data storage at the back end. Recent advances in Internet technology and the availability of suitable software environments offer the possibility of using different implementation strategies and tools. Figure 3 shows the architecture based on a Java 2 Enterprise Edition (J2EE) platform which is proposed for the Workflow Management Information System (WFMIS) under consideration.

When the web browser is used as the client interface, J2EE specifies the use of Servlets and Java Server Pages (JSP) technology to generate web pages. JavaBeans are used to encapsulate business logic and to connect to the data tier through drivers based on a Java Database Connectivity (JDBC) standard. In the present research, the Relational DBMS (RDBMS) is considered in response to the constraints specified by the client. In addition, the Simple Message Transfer Protocol (SMTP) is used to exchange messages between the application server and system users through an existing SMTP server.

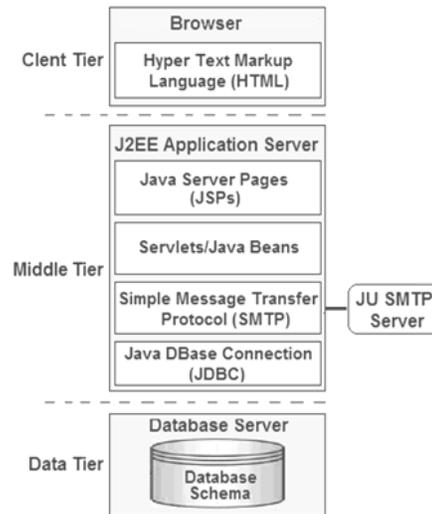


Fig. 3: Architecture of the WKFMIS

System Features: The features of the proposed workflow management system can be summarized as follows:

- Submission, distribution and monitoring of services over the web
- Different levels of access to system features
- Online generation of reports for past and present activities
- Real-time follow-up of work progress and timely notification of status
- Monitoring of personnel performance and optimising allocated jobs and time distribution
- Supports both Arabic and English languages
- Includes options for the job approval process that suits different styles of management

Methodology and scenarios: The object-oriented development process adopted in this study is started with an envisioning and requirement analysis phase where the system is divided into several subsystems. The process then proceeds with three iterative phases related to the subsystem's development; design, programming and evaluation. Each subsystem cycle is expected to take one to three weeks, as shown in Fig. 4. Finally, the system development process is completed by integrating the developed subsystems and evaluating the complete system prototype.

Process scenarios of the WKFMIS include four processes: user registration and management, order submission and management, e-mail notification and reporting. Of these, the latter two processes are newly added processes to the existing paper-based system.

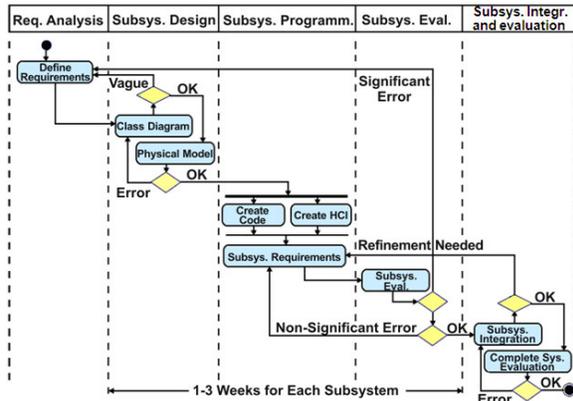


Fig. 4: System development phases

These scenarios are based on observations and information gathered from users in the working environment in FET. The Model-View-Control (MVC) architecture is widely used for interactive applications. This architecture divides the functionality among objects involved in maintaining and presenting data to minimize the degree of coupling between the objects^[13].

User registration and management: User registration starts by filling in the required information in a registration form. Once completed, the registration form is sent to the corresponding departmental manager for approval. Upon approval, the system automatically updates the user status, notifies the FET user via e-mail and sends the registration form to the administrator. The administrator then activates the user by updating the user's status in the database. Upon user activation, the system automatically notifies the newly registered user via e-mail. Other users (i.e., the corresponding departmental manager, lab. manager and the FET Dean) can also be notified depending on the configurations of the corresponding contacts profile. Figure 5 shows the MVC architecture of the user registration and management process.

Order Submission and management: Similar to the user registration and management process, the order submission and management process starts by filling in an order form. Once completed, the order form is sent to the corresponding departmental manager for approval. Upon approval, the system automatically updates the order status, notifies the FET user via e-mail and sends the order form to the lab manager. The lab manager then specifies a target completion time (in days) and forwards the order to one of the lab workers. The system also notifies the FET user, departmental manager and the assigned lab worker via e-mail.

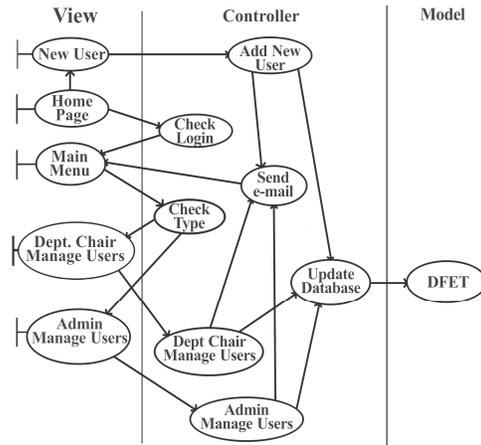


Fig. 5: MVC architecture of user registration process

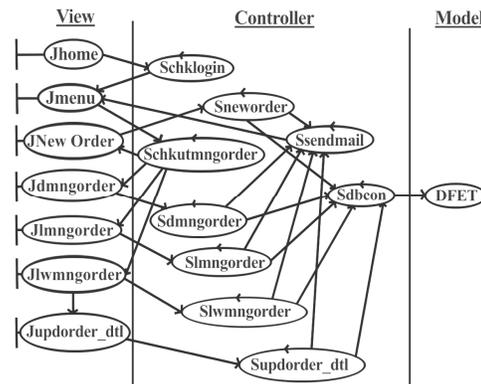


Fig. 6: MVC architecture of order submission process

The lab worker will then perform the required service and update the order status according to the work progress until completion. For each order status update (e.g., started, pending, overdue or completed), the system sends email messages to the corresponding users. Submitted orders can also be cancelled by the users who have ordered them.. The MVC architecture of the order submission and management process is shown in Fig. 6.

E-mail notification: This process aims at generating automatic e-mail notifications to all system users depending upon certain process events related to each user type. Table 1 summarises the events for which system users receive email notifications. The actor that initiates each event is designated with symbol (*) and the users to whom the messages are sent, are identified by shaded rectangles, as illustrated. The MVC architecture of the e-mail notification (contact) process is shown in Fig. 7.

Table 1 Events of e-mail notifications for users

	New user	FET user	Dept. manager	Lab. manager	Lab. worker	Sys. server	Admin.
New order	*						
New order approved by dept manager			*				
New order approved by lab manager				*			
New order by lab worker					*		
Order completed					*		
Order overdue						*	
Active order cancelled by FET user	*						
Active order cancelled by dept manager			*				
Active order cancelled by lab manager				*			
New user is registered	*						
New user approved by dept manager			*				
User is inactivated by admin.							*
Assigned to a new order				*			

Table 2: Types of users reports

	FET User	Dept Chair	Lab Manager	Lab Worker	Admin	Dean/Vice Dean
Personal orders reports						
Dept orders reports						
FET orders reports						
FET users reports						

Reporting process: This process aims at preparing and printing various types of reports. The contents of these reports vary depending on the user type. For example, FET users are allowed to prepare and print reports related to their own orders while the FET Dean can prepare reports on all orders and users existing in the system database. The contents of these reports can be filtered by status, department and time period. Table 2 summarises the types of reports which can be prepared by each user type. Unlike other processes, this process only retrieves, views and prints order-related and user-related information existing in the system database. Figure 8 shows the MVC architecture of the reports process.

UML design models: The object-oriented design process adopted in this work involves the development of several design models. These models which are essentially the system design represent a bridge between the system requirements and its implementation.

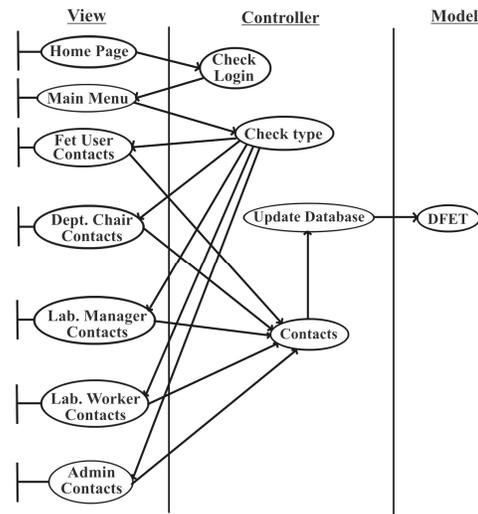


Fig. 7: MVC architecture of email notification

The Unified Modelling Language (UML) provides more than 10 different static and dynamic models that may be produced to document a certain system design^[13, 14]. For brevity, the models presented are limited to the static model (object-classes and their relationships), dynamic structure (the interactions between system objects, using collaboration diagram) and specification of object interfaces. The Entity Relationship (ER) diagram of the database design is also presented.

Static structure: The object-based design processes involve static models which describe the static structure of the system using classes and their relationships. A class diagram can be defined as a diagram that shows a set of classes, interfaces and their relationships.

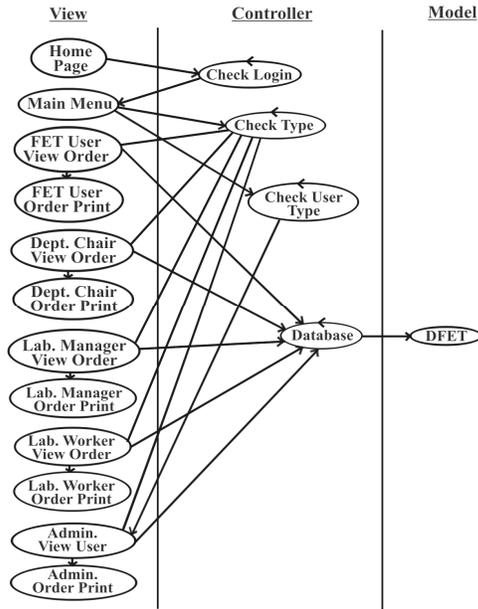


Fig. 8: MVC architecture of the reports process

The purpose of a class is to specify a classification of objects and to specify the features that characterize the structure and behaviour of those objects^[15]. The structural feature of the class is its attributes and the behavioural feature of the class is operations^[16]. Several types of relationships can exist between classes such as realization, dependency, aggregation, composite aggregation and generalization^[15,16]. However, the identified relationships between classes of the WKFMIS are based on the first three types of these relationships, as shown in Fig. 9.

Dynamic structure: The collaboration diagram is used to model the dynamic behaviour of the WKFMIS. Each message defines a particular communication between objects in an interaction. Developing a collaboration diagram for a web based application faces difficulty in modelling a JSP as a single class. This defeats the modeling objective in identifying the architecturally significant pieces within the model in their appropriate context. To overcome this difficulty, the UML extension mechanisms are used to partition each JSP into two conceptual elements^[15; 17], namely; client page and server page.

The relationship between these pages is also special and is defined as a <<Build>> relationship^[17] where the server page builds a client page. When a server page builds a client page, the result is an HTML stream that is sent to the browser through which the request was originated.

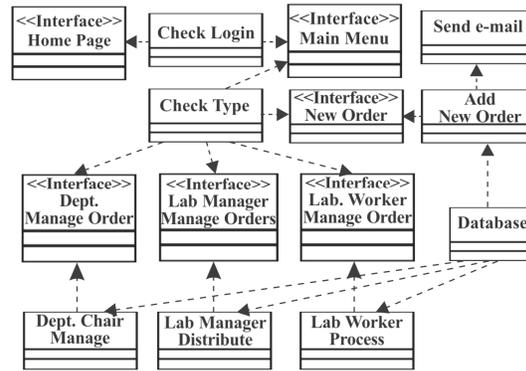


Fig. 9: Class diagram of the WKFMIS

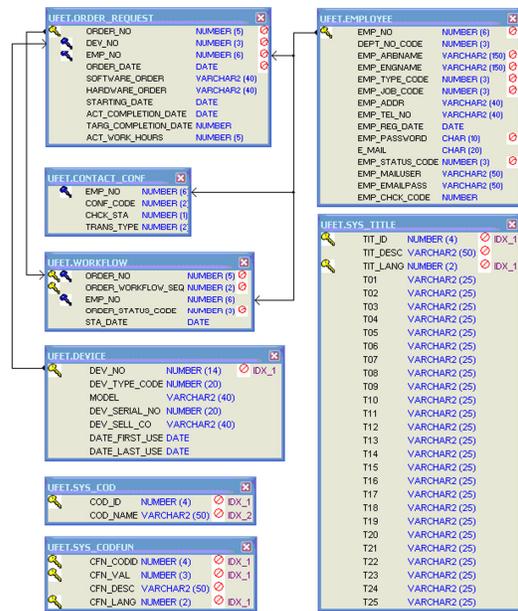


Fig. 10 ER diagram of system database

Database schema: As most processes in the WKFMIS are stateful and must survive a restart of the runtime engine, the business process implementation requires a persistent data store so that the process state is kept current in case of engine recovery. The data store is modelled as a normalized relational database schema, referred to in this study as “DFET”. DFET design is preceded by a stage of data gathering and analysis. This includes analysis of what kinds of information are expected to be retrieved, by system stakeholders, from the database. DFET is composed of five main tables and three utility tables. The main tables include: Employee, Order, Device, Workflow and Contact as shown in the Entity Relationship (ER) diagram of Fig. 10. The utility tables are only used for mapping titles from English to Arabic language (or vice versa)

and for mapping some system attributes to pre-defined codes with the aim of simplifying the coding process of the entire system.

MATERIALS AND METHODS

The system was developed and implemented in the Faculty of Engineering and Technology at the University of Jordan. The Database engine used was Oracle 9 running on a Windows 2003 server. The server also was running the J2EE platform which includes Java Server Pages (JSP), Java Beans/Servlets and the Java Dbase connection. Use was made of the University of Jordan mail server using the Simple Mail Transfer Protocol (SMTP) to send the email notifications.

RESULTS

The system is currently implemented and ready for full deployment. Our results indicate that several immediate improvements have become apparent. Table 3 shows extrapolated estimates for performance compared with the original paper-based system.

More than 25 connected pages (JSPs) are designed and implemented to display various types of information. These pages are categorized into six groups: common pages used by most or all users, FET user pages, department manager pages, lab manager pages, lab worker pages and administrator pages. For brevity, however, only samples of these pages are shown in Table 4.

DISCUSSION

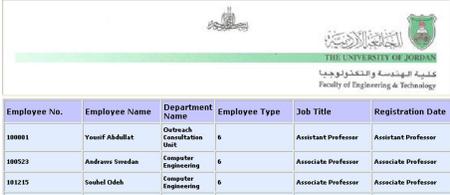
The use of the rapid prototyping approach in such a project proved very successful. Several brain storming sessions and feedback took place with all stakeholders. This helped a lot in fine tuning the design. Among the issues uncovered was the deep concern expressed by faculty users due to lack of timely status notification of their requests. This led to the incorporation of automated email status notification after each phase of the request process. Also, the reporting mechanism incorporated was highly appreciated as it allowed close monitoring of activities of all departments and staff and eased informative timely future planning decisions.

The use of the object oriented methodologies together with UML clearly helped in simplifying and structuring the design process. This led to quick implementation and testing which in turn allowed us to quickly gather feedback from users and relate that back to the design requirements. We discovered that this quick and timely activity on our part had a great impact on the administrative and working staff and their interest and collaboration.

Table 3: Performance of web vs. paper-based system

Process activity	Estimated processing time	
	Web-based system	Paper-based system
Place order	2 min	5 min
Approve order	0 - 1 min	15 min-4 h
Distribute order	2 min	15 min-1 h
Trace work progress	2 min	14 min-4 h
Prepare annual order report	5 min	1-2 days
Prepare annual user report	5 min	1-2 days

Table 4: Sample pages for the developed WKFMS

Page description	JSP Image
Main menu	
Order distribution report	
Admin report	

CONCLUSION

A web-based workflow management system is presented for the management, follow-up and status update of user requests for computer repair and maintenance at the University of Jordan. The detailed system architecture, models and realization is shown. The motivation and benefits for the adopted system are explored. It is seen that through the direct and constant interaction between system developers, users and administrators, it is possible to easily and quickly transform paper-based business processes with the help of current modelling and implementation technology such as UML and J2EE. The system is currently implemented with very encouraging results. The system design allows easy expansion; therefore, we aim to expand the system to include more automation options in the future. Some of the ideas being considered are: employing a computational intelligence approach for

workload distribution to lab workers thus letting the system automatically assigns jobs as they arrive. Also, we are considering introducing extra processes, also based on computational intelligence approaches, to give end users direct solutions to known problems taken from the system database, thus allowing the end users to solve their issues, if possible, without having to register a repair request. The generation of automated daily work schedules for lab workers is also being considered to streamline the workload process.

REFERENCES

1. Ellis, C.A. and G.J. Nutt, 1980. Office information systems and computer science. *Comput. Surveys*, 12: 27-60. <http://doi.acm.org/10.1145/356802.356805>
2. Whelpton, D. and D.K Cooke, 1990. Computer system for equipment management. *J. Biomed. Eng.*, 12: 248-252. DOI: 10.1016/0141-5425(90)90049-S
3. Mudie, D.C. and N.H. Chang, 1990. FAULTS: An equipment maintenance and repair system using a relational database. Proceedings of 9th International Symposium on Electronic Manufacturing Technology, Oct. 1-3, IEEE Xplore, Washington DC., USA., pp: 41-46. DOI: 10.1109/IEMT9.1990.114977
4. Avison, D.E. and G. Fitzgerald, 2003. Where now for development methodologies? *Commun. ACM.*, 46: 78-82. <http://doi.acm.org/10.1145/602421.602423>
5. Milosavljevic, G. and B. Perisic, 2003. Really rapid prototyping of large-scale business information systems. Proceedings 14th IEEE International Workshop on Rapid Systems Prototyping, June 9-11, IEEE Computer Society, San Diego, CA., USA., pp: 100-106. <http://portal.acm.org/citation.cfm?id=828276>
6. Krabbel, A., I. Wetzel and H. Zullighoven, 1997. On the inevitable intertwining of analysis and design: developing systems for complex cooperation. Proceeding of 2nd Conference Symposium on Designing Interactive Systems, Aug. 18-20, Amsterdam, The Netherlands, pp: 205-213. <http://doi.acm.org/10.1145/263552.263609>
7. Salimifard, K. and M. Wright, 2001. Petri net-based modeling of workflow systems: An overview. *Eur. J. Operat. Res.*, 134: 664-676. DOI: 10.1016/S0377-2217(00)00292-7
8. Crowe, M. and S. Kydd, 2001. Agents and suggestions in a web-based dynamic workflow model. *Automat. Construct.*, 10: 639- 643. DOI: 10.1016/S0926-5805(00)00072-8
9. Bin, S. and Y. GuoFu, 2002. The modeling method of workflow management system for agile enterprise. *Int. J. Comput. Integ. Manufact. Syst.*, 8: 105-110.
10. Kim, J., W. Suh and H. Lee, 2002. Document-based workflow modelling: A case-based reasoning approach. *Expert Syst. Appl.*, 23: 77-93. DOI: 10.1016/S0957-4174(02)00028-3
11. Lin, L., S. Zhong and Yu. Yao, 2005. Object-oriented distributed workflow management system. Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design, May 24-26, Coventry, UK., pp: 685-689. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1504174
12. Muhsin, Z., A.H. El-Mousa and I. Potts, 2007. Development of a web based database-centric workflow management system using rapid prototyping approach. Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, May 13-16, Amman, Jordan, pp: 92-99. DOI: 10.1109/AICCSA.2007.370869
13. Pressman, R.S., 2005. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, USA., ISBN: 0072853182.
14. Sommerville, I., 2004. *Software Engineering*. 7th Edn., Pearson Education Limited, UK., ISBN 0321210263, pp: 795.
15. Booch, G., J. Rumbaugh and I. Jacobson, 1999. *The Unified Modeling Language User Guide*. 1st Edn., Addison-Wesley Longman, Redwood City, CA., USA., pp: 482.
16. Object Management Group, 2007. *OMG unified modelling language specification: Version 1.5*. <http://www.omg.org/docs/formal/03-03-01.pdf>
17. Ahmed, K.Z. and C.E. Umrsh, 2002. *Developing Enterprise Java Application with J2EE and UML*. 1st Edn., Addison-Wesley, Reading, MA., ISBN: 0201738295, pp: 330.