I.J. Information Technology and Computer Science, **2013, 07, 36-42**
Published Online June 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2013.07.04

Evaluation of the Extended CBD Model: A Case Study using IMS Application

M. Rizwan Jameel Qureshi, Ahmed Barnawi
Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
E-mail: rmuhammd@kau.edu.sa, ambarnawi@kau.edu.sa

***Abstract*—** A large number of projects failed because of concentrating on developing new software. The Main drawbacks of new software development are more costly, and need maintenance. Reuse is solution of these problems that caused the widespread usage of object oriented (OO) development. Object oriented development is the backbone of component-based development (CBD). CBD facilitates reuse of the existing components (by customizing) with the new ones. Main advantages of reusable components are more reliable, saved time and reduced cost. CBD approach is different from traditional/new software development. Several models have been proposed for traditional software development such as Waterfall, Rapid Application Development (RAD), Evolutionary, Rational Unified Process (RUP) and agile. Whereas the popular CBD models, for customized development, are V, Y, Umbrella and W. Almost all software development companies, both the new software and customized software have to be developed. Therefore majority of the software development companies face problem to select and implement an appropriate process model for the both kinds of developments. To address this problem, the existing RUP and CBD models do work to some extent but the authors intend to propose an extended CBD model that it equally offers its benefits for new and customized developments. The experimental data is taken from a case study to develop IP Multimedia Sub System (IMS)-based examination application using iPhone to evaluate the proposed model. The results provide evidence that the extended CBD model can be equally beneficial for the development of both new and customization components for IMS-based applications.

***Index Terms*—** Component-Based Development for Customized Projects, Software Development Life Cycle, Reusability, Repository, Case Study, Quality

I. Introduction

Component-based software engineering (CBSE) is used to develop/assemble software from existing components. Software developed using components is called component ware [1]. Component based development idea was in consideration from last many years [2]. Engineering and scientific libraries are available with the state of art tools to reuse the previously developed functions [1]. Several models are proposed for structured development such as Waterfall, Prototype, Rapid Application development and Spiral [1]. The traditional models are supportive to new development from the scratch and there is no reference for reuse/customization of existing components. Due to this problem, the traditional models are not suitable for software industry where reuse (of design and code) is an essential requirement [3]. A model named Rational Unified Process (RUP) was introduced to remove the shortcoming of traditional models [1].

RUP model incorporated many characteristics of Spiral model focusing on pure object-oriented development [1]. Inception, Elaboration, Construction and Transition were the main phases of RUP model [1]. Construction phase supported reusability of existing classes. The main shortcomings of RUP model were [1]:

- time consuming;
- costly;
- excessive emphasis on documentation rather than on development;
- heavily relied on the expertise because it must be tailored for each project;
- complex in nature to implement for the new and customized software development projects.

There were numerous process models introduced to remove the shortcomings of RUP model such as CBD, Agile, Cleanroom Software Engineering, Agile subset of the RUP practices, Microsoft Solutions Framework (MSF), Personal Software Process (PSP) and User Case Responsibility Driven Analysis and Design (URDAD) [1][4]. CBD models were the process models mainly among all models to remove the shortcomings of RUP model and these models had gained great popularity in software development community from last many years.

Software industry has to face a problem to select a suitable model that supports equally for the development of customized and new components to deliver products timely. Traditional models do not match the current needs of software industry to reuse previously developed components [3]. With that less empirical studies are available for the implementation of existing CBD models because of a gap between academia and industry. Study in [3][5] shows that software industry is rarely implemented the proposed CBD models. The main reason is extensive heavyweight proposed processes of existing CBD models need more effort and time. Software industry has to bear an immense amount of stress to deliver products timely without sacrificing quality. This paper focuses on extending CBD model for customized and new developments to meet the industrial demand.

Rest of the paper is organized as follows; Section 2 covers the related that is used to identify the research problem subsequently in sub section 2.1. An extended CBD model is presented in Section 3. Section 4 presents experiment and analysis of the proposed extended CBD model using case study.

II. Related Work

A software-cycle model [6] was proposed for reuse and reengineering. The proposed model suggested five stages to reuse a component [6].

- Analysis of existing programs to sort components to be reused.
- Reengineering to eliminate domain specific troubles.
- Saving reusable components in the repository.
- Construction of independent status components with a reuse approach to store in a repository.
- Reuse components to develop new programs.

It was not a complete process model for CBD but main focus of their research was on reuse activities [6]. The drawback of this CBD model was removed by proposing an improved CBD model [7]. The main phases of improved CBD model were ‘Component Analysis’, ‘Architectural Design’, ‘Component Brokerage’, ‘Component Production’ and ‘Component Integration’ phases [7]. The improved CBD model was a modification of Waterfall model integrated with these phases. There were some limitations in the improved CBD model. The improved CBD model was not suitable for commercial applications because of the verification of phases repeatedly. It was more time and cost consuming process model and suitable only when comprehensive or stable requirements were available to software engineering team.

Lycett and Giaglis [8] discussed Discount Cash Flows (DCF), Net Present Value (NPV), Return on Investment (ROI), Internal Rate of Return (IRR), SESAME and cost benefit analysis sheet (CBA) but without providing any facts and figures. They proposed a content, context and process (CCP) analysis to evaluate a component for reuse. CCP analysis is not practical from implementation point of view because it is very subjective for selection and evaluation of a component for reuse whereas Constructive Cost Model II in [9] is far better to calculate cost of new systems to be developed by new and reusable components.

Package based software development is a popular research area [10]. de Jonge [10] concentrates on source tree composition technique for software configuration management of more than one system. This is a problem in terms of management of reusable components in a repository. There is no discussion about crosscutting development of components and their integration in multiple systems.

The extent of software reuse depends upon the reuse strategy followed [11]. The five strategies were ad-hoc reuse with high reuse potential, uncoordinated reuse attempt with low reuse potential, uncoordinated reuse attempt with high reuse potential, systematic reuse with low management support and systematic reuse with high management support. Main objective of their research is to classify reuse strategies so that development groups can get benefit and achieve success to complete software projects. Marcus et al. [11] support the last strategy i.e., systematic reuse with high management support, but this scheme needs highly detailed analysis by gathering of data from various software houses working in different geographical locations, before reaching a conclusion.

Selection of reusable components is important to improve productivity of component-based software [12]. Jihyun et al. [12] proposed a Component Repository for facilitating Enterprise Java Beans (EJB) Component Reuse (CRECOR) to store and manages the reusable components. Working on the reusable components with repository (software) has many benefits such as specification viewing, adapting, testing and deploying.

The repository, proposed by Jihyun et al. [12], does not have a version control and change control functions to manage different versions of reusable components. Version and change control are important functions of a repository to manage and update different versions of a component.

Poorly gathered SW requirements could fail a SW project [13]. It was because of natural drawback in requirements determination methods. An approach was proposed to construct SW by categorizing components in a knowledge base. Existing components were recognized, chosen and integrated in a newly developed SW by using the knowledge base. Different classification schemes to reuse artifacts had been discussed as well. The objective of this paper was to ease requirements gathering using knowledge base but the paper lacked in suggesting a comprehensive process model for CBD. An attempt was made in 2004 to remove the limitations of existing CBD models [14]. An incremental method was presented for distributed CBD [14]. It was based on two phases. The first phase composed of gathering requirements of the problem domain and construction of employable components in object-oriented (OO) language. These employable components were stored in a repository. Software engineers looked up MVCASE tool to select the necessary components to develop the SW system in the second phase. There were two main weaknesses of this model. It was not a complete model and a use of a specific tool was the requirement of this process model. The efficiency of existing CBD models was improved [15].

The four-stage component-based development process model was very complex for implementation [15]. The core objective of their research was to integrate off-the-shelf components with the newly developed components rather than in house development. Repository had not been used here. Repository is an important element of CBD process to classify, select and retrieve components for customized development [16]. Y-model was proposed in to remove the limitations of existing CBD models [17]. Y-model was anticipated to support CBD using object oriented (OO) construction [17]. According to Luiz [17], the main phases of Y-model were 'Domain Engineering', 'System Analysis', Design and Implementation. The major problem of their model was the selection of reusable components during the design phase. The selection of reusable components should be during the analysis phase. Therefore, analyst could estimate the cost, schedule and effort to develop and integrate the components.

Haddad [18] is of the view that software organizations have to invest huge sums of money to start successful reuse methodology and it's a barrier for them. Haddad believes that core of reuse is source code and proposed an integrated approach for component-based development to support domain specific components. A wrapper interface mechanism is also discussed to manage and control the interface between or among integrated collection of reusable components. The objective of this research is to develop benchmarks for software organizations so that these begin reuse practices by emphasizing mainly on programming effort and not on management and operational issues. Focus is development of domain specific reusable components and not construction of reusable components of different domains of concern [18]. This problem can be handled through software engineering for adaptive and self-managing systems.

2.1 Research Problem

A number of attempts, V, Y, Umbrella and W, have been made to propose CBD models in the last many years [2]. However most of them are not in use due to extensive heavyweight processes that need more time and cost making them unsuitable for software industry [3]. Further, the proposed CBD models lack in providing empirical evaluations for new and customized developments [3]. So, there is a need for a simple and light extended CBD model to provide an empirical evidence of a study in an industrial context. The research question therefore becomes:

"How to extend CBD process model for the development of new and customized developments?"

III. The Proposed Extended CBD Model

In this section, the proposed tailored CBD methodology is presented. It is an extended form of the classical CBD methodologies that is expected to equally suitable for new and customized development by removing the limitations of existing CBD models. Fig. 1 shows the proposed extended CBD model. The main phases of the improved CBD are 'Project Planning', 'Analysis, Selection and Risk Management', 'Adaptation, Engineering & Integration' and 'Testing'. A detailed description of the phases of the extended CBD model is as follows.

'Project Planning' Phase- Project planning is one of the project drivers and it provides a systematic or pragmatic approach to direct and complete a project. Project planning plays an important role for the success or failure of a project. A software project will be deteriorated after few years if the project planning is not properly made and documented [1]. Project specification document is prepared during the 'Project Planning' phase. The main activities are defining the project scope and its alternatives, feasibility assessments, dividing the project into manageable tasks, estimating resources, developing a schedule, identifying major risks and its assessments and use case modeling.

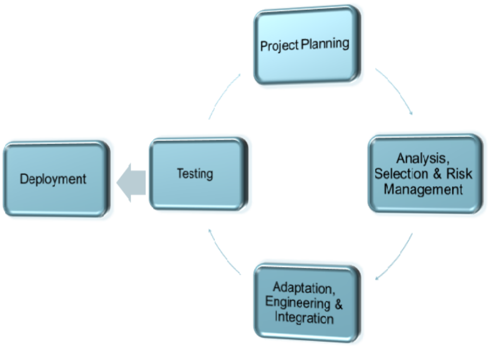

Project Planning
Analysis, Selection & Risk Management
Adaptation, Engineering & Integration
Testing
Deployment

Fig. 1: The Proposed Extended CBD Methodology

'Analysis, Selection and Risk Management' Phase- Analysis phase is initiated if the customer approves the proposal. The name of analysis phase is 'Analysis, Selection and Risk Management' phase in the new framework. This is the phase where an analyst gathers the detailed requirements of the system to be developed. A domain analysis is performed to find a suitable architecture for the application to be developed [1]. An architectural model of application is developed that enables software engineer to:

- evaluates efficiency of design;
- judge options of design;

- minimize potential threats coupled with software development [1].

'Analysis, Selection & Risk Management' is the phase where an analyst tries to identify and select those components that can be reused from the components repository. The selection of reusable components is important to improve productivity of component-based software development. The repository has a profound effect on the CBD model to be proposed to store and manage reusable components. Main benefits achieved, while working on the reusable components having a repository are classification, searching, modification, testing, implementation, version control & change control, update and consistent documentation. There could be one or more than one repository to select and retrieve components [1]. The main benefit of more than one repository is that repositories can be developed and categorized on the basis of specific component domains.

Risks about new and existing components are also evaluated and managed. Reusable components need qualification, adaptation and composition. Component qualification makes sure that the selected component will:

- execute the desired functionality;
- integrate easily into the structural design of new application;
- demonstrate the quality attributes (e.g., reliability, performance, usability) [1].

The properties, behavior and relationships among components are identified. Core objective of this phase is to reuse components as maximum as possible, rather than reinventing the wheel. It will also improve productivity and efficiency of software engineers.

'Adaptation, Engineering & Integration' Phase- The reusable components are customized according to the requirements of the new system to be developed and tested during 'Adaptation, Engineering & Integration' phase. A common component adaptation technique called component wrapping is used if programmer is using black box components [1]. Software engineering methods are applied to design and develop new components for those requirements which cannot be fulfilled from already developed components. The customized and new components are integrated into system.

Testing Phase- The new components are designed, developed and tested on unit basis. Integration and system tests of the newly developed and of the reused components are performed. A customer is requested to evaluate and verify software, whether it meets his/her requirements or not during the testing phase. The software is ready to deploy at customer site. The main activities of deployment are installation, training and security.

3.1 The Proposed Extended CBD Model Main Activities

The main activities of are domain engineering, structural modeling and functional characterization. The intent of domain engineering is to identify, construct and catalog set of software components that are implementable to existing and future software. The overall objective is to set up mechanisms that facilitate the SW engineers to share and reuse the components to develop new and existing systems using a repository. These questions are answered for identification of reusable components.

- Is component functionality required on future implementations?
- How common is the component's function within the domain?
- Is the design optimized enough for the next implementation?
- Is the component reusable in many implementations with only minor changes?
- Is reuse through modification feasible?
- Can a non reusable component be decomposed to result in a reusable component?
- How valid is component decomposition for reuse?

When new SW 'W' is to be built within the application domain, it is difficult to determine whether a potentially reusable component is applicable or not. It is necessary to define a set of domain characteristics that are shared by all SW within a domain to facilitate determination. A domain characteristic defines some generic attributes of all products that exist within the domain. For example, generic attributes might include:

- The importance of safety/reliability;
- Programming language;
- Concurrency in processing.

A set of domain characteristics for a reusable component can be represented as {Dp} where each item 'Dpi' in the set represents a specific domain characteristic. The value assigned to 'Dpi' represented as scale ranging from 1 to 5. High value indicates more relevance of the characteristic for component 'p'. A typical scale might be:

- Not relevant to reuse;
- Relevant only under unusual conditions;
- Relevant-the component needs major modification;
- Clearly relevant-the component needs minor modification;
- Complete matching- the component needs no modification.

A comparison is made between Dpi and Dwi to determine that how many components can be reused. When domain engineering is applied, the analyst looks for repeating patterns in the applications that reside within a domain. Structural Modeling is a pattern based domain engineering whose objective is that every application domain has repeating patterns (of function, data, and behavior) which can be reused. Every domain can be characterized by a structural model e.g., financial application, embedded SW, health care system, educational systems. Therefore the structural model can be and should be reused across applications within the domain. A structure point is a unique construct within a structural model. It is possible to define generic structure points that cover a number of application domains.

- Application front end-the GUI including menus, panels and input handling
- Database-the repository of all objects relevant to the application domain
- Computational engine-application logic (processing and business logic)
- Reporting functions
- Application editor-that is used to customize the application to the needs of specific users.

IV. Experiment and Analysis

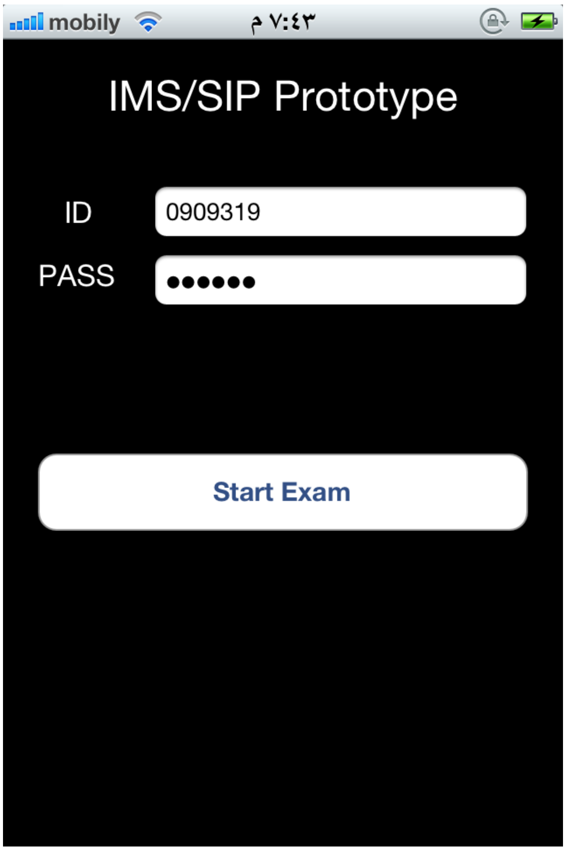
The traditional methodology used paper-based examination to evaluate the students. The paper-based examination has many drawbacks such as time and cost consuming. The instructors have to consume too much time to prepare question paper and marking of answer sheets in each semester and for each course. Finding a suitable location could be another challenge for the organization. Due to these reasons, a case study is conducted to change the old methods of conducting examinations with the newer one by using the state of art technology i.e., IMS based mass mobile examination system. It can solve most of the problems associated with the paper-based traditional examination system.

The main objective of this research is to build a new examination system that is more efficient than the traditional examination system using the extended CBD model. Moreover, there are other objectives to achieve in this research as follows.

1. Enhance the acceptability, usability, and security.
2. Increase the flexibility of learning.
3. Encourage and support information technology environments.
4. Improve the academic learning in the Kingdom of Saudi Arabia.
5. To develop a mobile application environment so, the students can use it to do their exams on it.
6. Devolve a new examination technique of exams that is more efficient than the traditional way.

4.1 Materials, Tools and Research Methodology

The main resources are used to conduct the case study are iPhone, IMS Server and computers. The tool is Objective-C programming language to develop client side. For the server side, MYSQL database and JAVA programming language are used. Student entered his KAU ID and Password and the application checked the authentication for the exam. If not, the application does not allow him to download the questions. The client application received the questions to submit answers. The exam answers are sent to the IMS Server for the evaluation purpose. The marks, mistakes and right answers are shown to the student in his iPhone. IMS server manages exam questions, time and students who are taking exam.

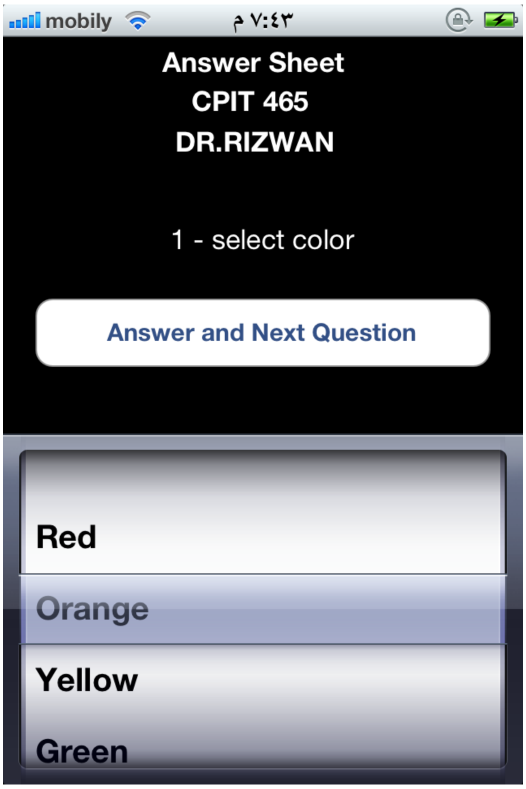


ID 0909319

PASS ●●●●●●

Fig. 2: interface to authenticate student

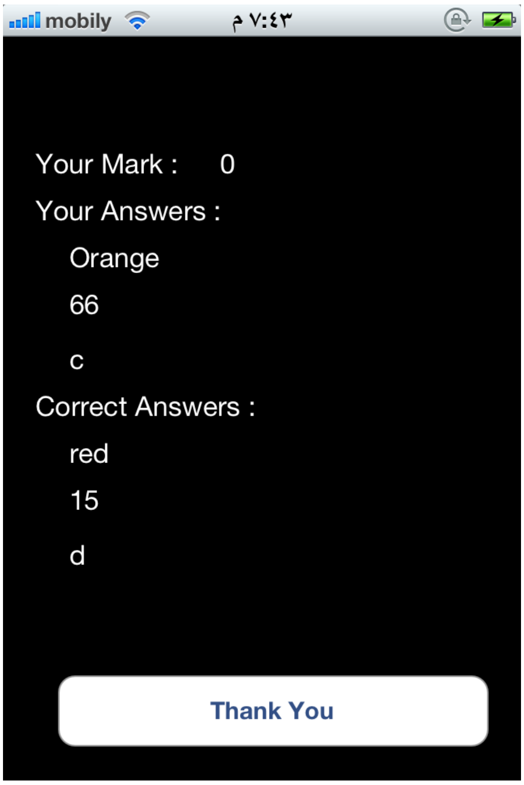
As results, some snap shots for the IMS based application are shown in fig. 2 to fig.4. The application started with authentication and the exam question is displayed to student in fig. 2 and fig.3. After answering the all questions, the exam is sent to IMS server for the evaluation purpose and the results are displayed on the iPhone in terms of marks, mistake and right answers in fig. 4.



Answer Sheet
CPIT 465
DR.RIZWAN

1 - select color

Fig. 3: interface to display question to students



Your Mark : 0

Your Answers :

Orange

66

c

Correct Answers :

red

15

d

Fig. 4: interface to display results

The successful implementation of IMS-based application justifies the proposal of the extended CBD model.

V. Conclusion

The CBD models are a desirable replacement of traditional heavyweight models for the software development companies at the current time to support reuse. Software industry has to deal with new development as well. Software development companies face it difficult to select a suitable model that supports equally for the both types of development. Traditional and existing CBD models are not fulfilling the need of software industry due to their heavyweight and complex processes requiring more time and effort and less support for customized and new development. Studies show that there is a need of extended CBD model that meet the industrial demand in major. On the face of it, an extended CBD model is proposed by addressing the limitations of existing CBD models for new and customized developments. Suitability of the proposed extended CBD model is evaluated by a case study to develop an IMS-based iPhone mobile application for mass examination system.

From the evaluation, the authors have found evidences that the proposal of extended CBD model is suitable for new and customized developments. However, the need for the statistical validation of the extended CBD model will be addressed in days to come by comparing with future releases/versions.

References

[1] Pressman R S. Software Engineering, McGraw Hill, USA, 2010.

[2] Crnkovic I, Larsson S, Chaudron M. Component-Based Development Process and Component Lifecycle [J]. Computing and Information Technology, 2005,13(4):321-327.

[3] Kaur K, Singh H. Candidate Process Models for Component Based Software Development [J]. Software Engineering, 2009,4(1):16-29.

[4] Klopper R, Gruner S, Kourie D G. Assessment of a Framework to Compare Software Development Methodologies [C]. In: Proceedings of Annual Conf. of the South African Institute of Computer Scientists and Information Technologists (SAICSIT' 07), 2007:56-65.

[5] Graham I. Object Oriented Methods, Principles and Practices, Addison Wesley, USA, 2000.

[6] John W B, Victor R B. The Software-Cycle Model for Reengineering and Reuse [C], In: Proceedings of the conference on TRI-Ada '91: today's accomplishments; tomorrow's expectations (TRI-Ada '91), 1991: 267-28.

[7] Ning J Q. A Component-Based Software Development Model [C]. In: Proceedings of 20th Conference on Computer Software and Applications, 1996: 389-394.

[8] Lycett M. Giaglis G M. Component Based Information Systems: Towards a Framework for Evaluation [C]. In: Proceedings of the 33rd Annual International Conference System Sciences, 2001: 10

[9] Boehm B, Chris Abts A B, Sunita C, Bradford C, Ellis H, Ray M, Donald R, Bert S. Software Cost Estimation with COCOMO II, Pearson Education, USA, 2000.

[10] De Jonge M. Package-Based Software Development [C]. In: Proceedings of 29th EUROMICRO Conference on New Waves in System Architecture, 2003: 76-78.

[11] Marcus A R, Keven J D, Uday R K, Nader N. Strategies for Software Reuse: A Principal Component Analysis of Reuse Practices [J]. IEEE Transactions on Software Engineering, 2003,29(9):825-837.

[12] Jihyun L, Jinsam K, Gyu-Sang S. Facilitating Reuse of Software Components using Repository Technology [C]. In: Proceedings of 10th Asia-Pacific Conf. Software Engineering, 2003: 136-142.

[13] Hemant J, Padmal V, Fatemah M Z. An Assessment Model for Requirements Identification in Component-Based Software Development [J]. The DATABASE for Advances in Information Systems, 2003,34(4):48-63.

[14] De Almeida E S, Alvaro A, Lucredio D, Do Prado A F, Trevelin L C. Distributed Component-Based Software Development: An Incremental Approach [C]. In: Proceedings of 28th Annual International Conference on Computer Software and Applications, 2004: 4-9.

[15] Hutchinson J, Kotonya G, Sommerville I, Hall S. A Service Model for Component-Based Development [C]. In: Proceedings of 30th EUROMICRO Conference, 2004: 162-169.

[16] Kaur K, Kaur P, Bedi J, Singh H. Towards a suitable and systematic Approach for component base software development [C]. In: Proceedings of World Academy of Science, Engineering and Technology, 2007: 190-193.

[17] Luiz F C. Y: A New Component-Based Software Lifecycle [J]. Computer Science, 2005,1(1):76-82.

[18] Haddad H M. Integrated Collections: Approach to Software Component Reuse [C]. In: Proceedings of 3rd International Conference on Information Technology: New Generations, 2006: 28-33.

Authors' Profiles

M. Rizwan Jameel Qureshi: Assistant Professor of Information Technology, Faculty of Computing & Information Technology, King Abdulaziz University, major in software engineering and database management systems.

Ahmed Barnawi: Assistant Professor of Information Technology, Faculty of Computing & Information Technology, King Abdulaziz University, interested in software engineering and networks.

How to cite this paper: M. Rizwan Jameel Qureshi, Ahmed Barnawi,"Evaluation of the Extended CBD Model: A Case Study using IMS Application", International Journal of Information Technology and Computer Science(IJITCS), vol.5, no.7, pp.36-42, 2013. DOI: 10.5815/ijitcs.2013.07.04