University of Sindh Journal of Information and Communication Technology (USJICT)
Volume 2, Issue 2, April 2018
ISSN-E: 2523-1235, ISSN-P: 2521-5582
Website: http://sujo.usindh.edu.pk/index.php/USJICT/

Formal Approach for UML Components based Development Profile

Arifa Bhutto[1], D.M. Akber Hussain[1], Imran Anwar Ujjan[2], Mehran Syed[2]

[1]Department of Energy Technology, Aalborg University, Denmark
[2]Institute of Information and Communication Technology, University of Sindh, Jamshoro
abh@et.aau.dk, akh@et.aau.dk, iujan@yahoo.com, Mehransyed92@gmail.com

Abstract - Component based software development CBSD is a very complex and difficult process, specifically modelling and integration of various components to make one complete running large system. However, various types of software modelling process are available for development of CBSD. In this research, we focus to describe the concept, principles and architecture of CBSD, further we explain the life cycle of development phases of components based software engineering process. We describe the Unified Modelling Language (UML) for the architectural description of the CBSD system using School Management System case study to adopt our research approach.

Keywords: Components based software engineering, Engineering Process, Software development process

I. INTRODUCTION

The software engineering is integral part of the various industrial engineering domain such as communications, management, medical and control engineering system. Dew to the complex structure of the industrial design we need the efficient and reliable software engineering process. However, software engineering currently main focus on how to design and develop cost effective, reliable and on time development process which meet the criteria of the requirements having no failure perspective. Components-based software engineering (CBSE) model is new design and development of software applications which meet the criteria of industrial design process effectively and efficiently. CBSE process most popular because of the lesser time for development as well as decreases the cost of the software application [1]. Since many years' components based development process widely used in different domains and gain success too. CBSE process is effectively and more reliable in various domain such as desktop, graphics, web-based applications. In these above domains various component technologies are used to develop the system like a COM, .NET, EJB, J2EE are used [9][4].

However still always need for searching a better and latest process for new requirements which fulfil the currently demand of industrial process for reliable and cost-effective and efficient software development [7]. Our research paper is based on the understanding of the CBSE development models by various researchers, we further define Components based software engineering process CBSE according to our need and requirement the new practical way for development process which is based on the main methodologies of CBSE modelling process using UML notations. In this paper we define the major CBSD process and focus on UML modelling approach to define the practical approach using the case study of School Management System which design model based on the UML based Components methodology.

II. Differentiation between traditional development process and component based software development process.

Development of the software product start from selection of appropriate lifecycle in which we design, develop and test the project till the completion of the project. However, there is various software engineering process model are defining major activities of development for software, like waterfall model, incremental model, iterative model, evolutionary model, V model, screw model, RAID model [9].

Above all development models are define the basic structure for the software engineering process for the software applications, but as we know above are has few weaknesses and can't fit in components based development methodologies in table 5. We describe the basic difference between CBSD and all above traditional software development methodologies [5].

TABLE 1 Difference between Traditional and CBSD process

Components Based Software Engineering Task	**Traditional way of Software Engineering Task**
Mostly developed from the pre-existing components.	System developed from the start from scratch.
All related components are selected and developed the new system.	System newly developed based on the requirement of the systems.
More emphasis on the components selection and evaluation.	Traditional way have no such selection phase is available for the pre design system.

More time is required for the selection, testing, and verification of the components	More time is required for the system development.
Reusability is the major concern area in the components based development.	Reusability is not required in the traditional system.

A. CBSD Lifecycle models.

The CBSE process is to purchase not to develops [4]. CBSE purpose to realizing how to minimize the long waited software development process in the quick and effective software development process [4].

CBSE development process not only build the new components but integrated all components together to build the new system. Mostly research based on to searching already build components that can be integrated with new design system to minimize the time, cost and reliability of the system. The mainly CBSD process which following researchers define like Twin Peak model use of COTS product but it has little detail guidelines are missing shown in Figure 1. Summerville define the sequential approach for CBSD but Ivica Crnkovic developed the V model which shown in Figure 2 V model for components development process [9], Luiz Fernando Capretz[4] define new way of CBSE life cycle shown in Fig 3 as Y model which described the combination of linear and V model.

However, above all authors models define the major actives of the components development process which we explicitly customized by the important activities related with the CBSD process, following are the important components development process phases are define.

- Detail Requirements Analysis and Design
- Software Architecture and selection Process
- Components selection and identification
- Software System Integration and development
- Software System Testing and verification
- Software System Implementation and Maintenance

However, we define more appropriate and practical process for CBSD in Figure 4 which we used in our case study. Our main objective of the research to define components based system as architectural, modular top-level design of the system. In this perspective we choose a UML 2.0 as Components based development models for our case study using that we described the how component based design developed by using UML notations.

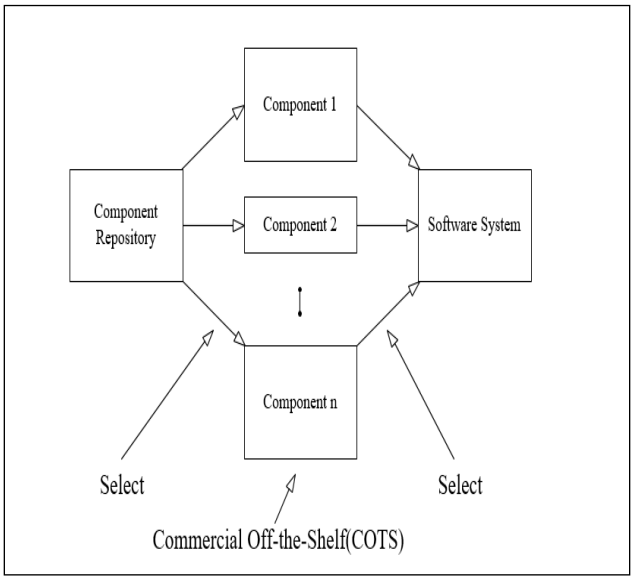


Figure.1: COTS design process model for CBSD

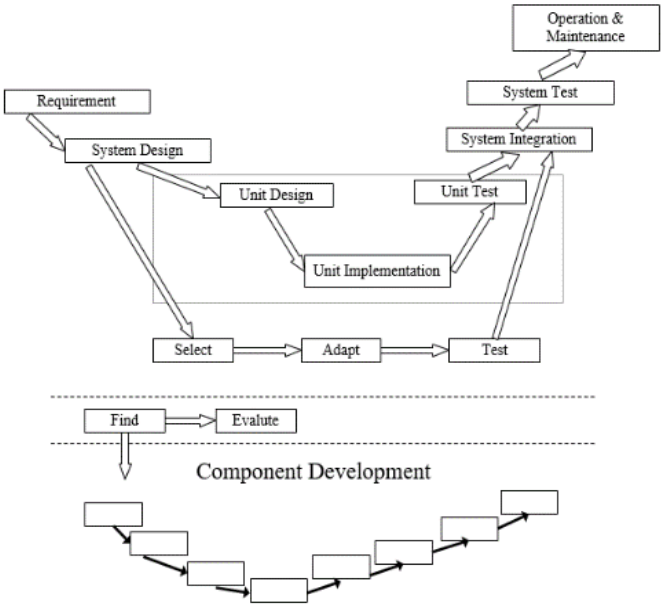


Figure 2: V Model for CBSD process [4,5,9]

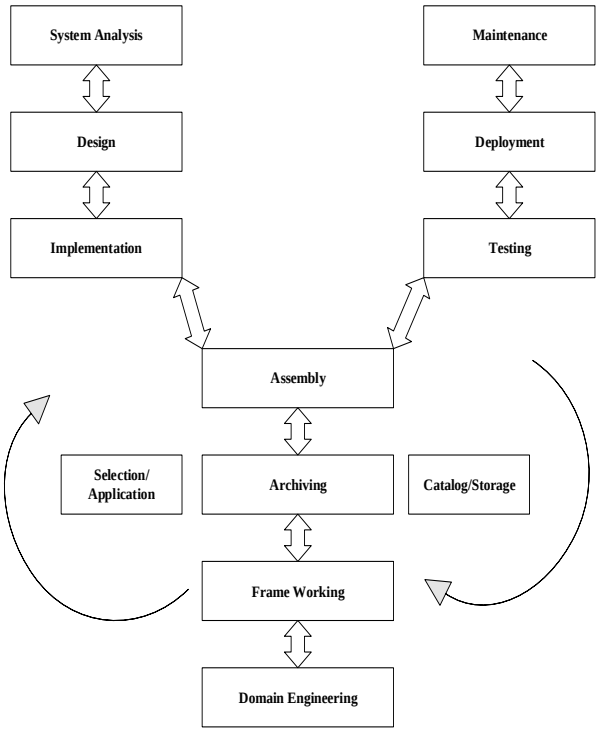


Figure 3: Y Model for CBSD Process [4,5.9]

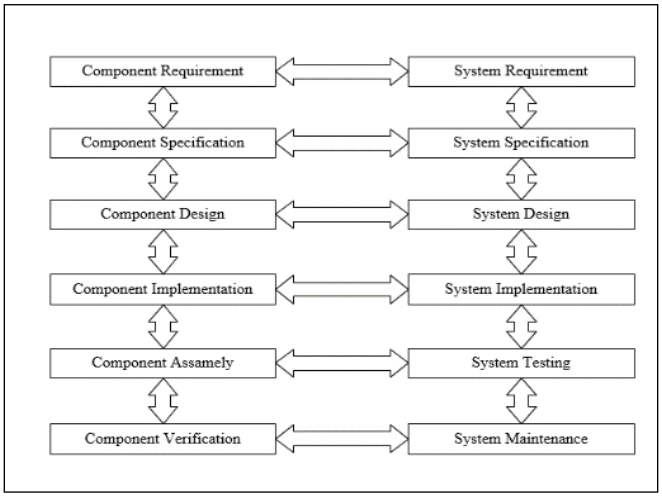


Figure 4: Practicable Approach of Component Based System Development Process Life cycle

III. SYNTAX OF COMPONENT

Syntax of components selection start from the requirement and design where we have to choose the architectural design application methods like UML or ADL for design of the components and represent the full complete model of the system and integration of all related components and their relationships. Second level start with development of components in that level we choose various programming language platform but most popular is object oriented environment like C#, ETB, .NET. Similar way we know all components are developed separately and for that intergradation and mapping level is third level where we have to choose the integration and development environment like COM, .NET and web services described in table 2. However, in our research we focused on to development of architecture of system using the UML 2.0 language, we show how components notation are development with example in fig 5.In UML components notation described in rectangle box like "Expensive" and "Payment" shows in Figure 5 there is always two major parts of components "Provider" and "Required" that shows interface notation of provider in circle "Payment" and required half circle "Transaction".

TABLE 2: Syntax of Components

Syntax of Component	
Object Oriented Languages	C#, ETB
IDL mapping	COM, .NET, web services
Architecture design applications	UML, ADL

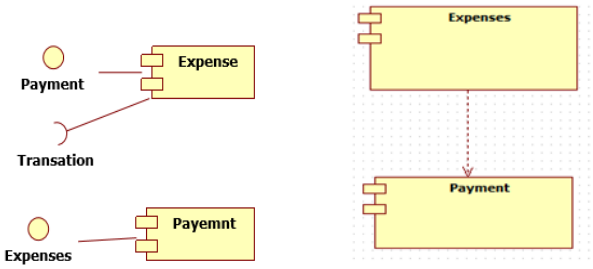
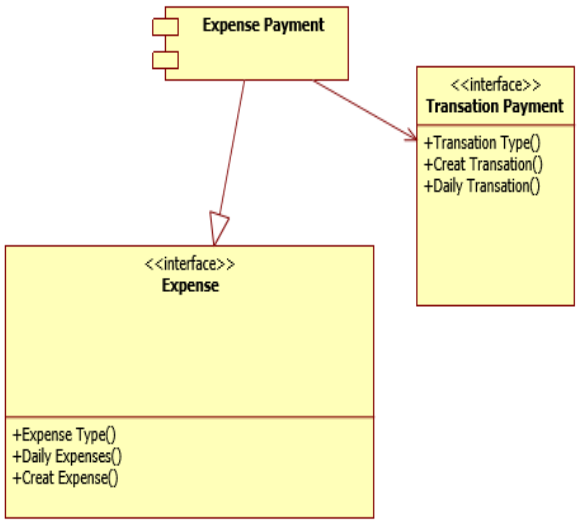


Figure 5: Component Diagram with required and service provider

A. Semantics of Components

In UML 2.0 semantics of components define internal behaviour as a provider and required interface with statistic and dynamic semantics. The internal architecture of component is in form of class, datatypes and operations. In Figure 7 we show the Expense Payment component including internal interface of provider and required interface of class, which shows data types and operations of the component including required and provider interface.



Syntax of Component	Semantic of component	System Developing
Object Oriented Programming .Net Framework	Object	DLL Classes Entity Frame works
Architectural Language UML	Use Cases, Component Diagram	Students Reports Accounts Salary Attendance

Figure 6: Semantic of Components in UM 2.0.

B. Component Composition

Component development and design usually stand alone, but all independent components always integrated with the larger system to make one complete software application. A component composition is based on the external view by means of interface, properties and operations. In our case study, we design the complete interface of the various models and their relationships we show complete model in Figure 7

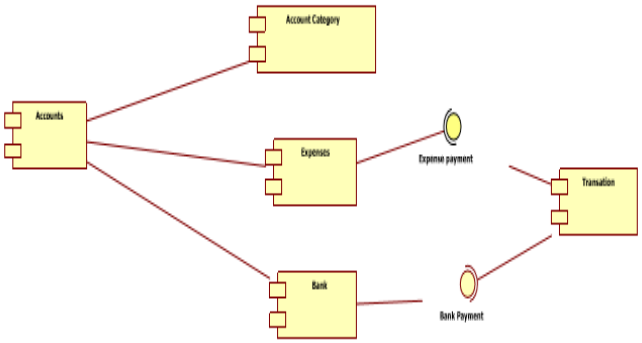


Figure 7: Components composition example in UML 2.0

IV. OUR PROPOSED DESIGN APPROACH

The proposed way of Components development process defines in Fig 4, which is based on major factor to be considered for design of Components based development process. The main factor for components based design process is components Interface, Usage Information, Deployment and Use define in Figure 8 major key area of CBD process. However, we map our we map SMS case study according to the proposed system of CBSD in table 3 shows the syntax of components, semantic of
components and system developing tools and languages used for the SMS.

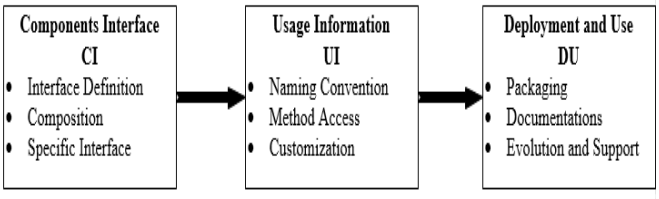


Figure 8. Components composition in UML

Table 2: Mapping of SMS according to the CBSD.

V. Case Study of School Management System (SMS)

School ERP System is a complete solution for a school management system for the admin work. The SMS has been developed according to the CBSD process. We described here in detail the architectural level of the SMS application. The application has various components like maintain student, fees, fee collection, employee, payroll management, library management.
The system architecture design shows in Figure 8 where all components shows relationships including required and provider interface.
The main features of the system design by the CBD Process are following:

- Student admission module
- Class distribution and timetable module
- Employee management module
- Payroll management module
- Accounts management module
- Generating Reports

The above all module has been design in UML2.0 Component model design in Figure 9 in detail including with there interface and internal architecture.

VI. CONCLUSION

To make a system complete by integrating various components Component based software development (CBD) process for modelling and integration is used. We focused on efficiency and reality of soft development process. There are various ways of modelling is available for CBSD. We have tried to describe the architecture, concept and principles of CBSD that are described in this paper. Lifecycle of the development process of components is also explained keeping an eye on the CBSD.
The research has followed the unified modelling language UML for the architectural description of the CBSD system having a case study of school management system. The main features of the system design by CBSD covered are Student management module, Class management module, Employee management modulee, Payroll management module, Accounts management module.

REFERENCES

[1] L. Bassi, C. Secchi, M. Bonfé, and C. Fantuzzi, “A SysML-based methodology for manufacturing machinery modeling and design,” *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1049–1062, 2011.

[2] A. Khanjani, “Comparison between Four Software

Engineering Approaches : Component Based Software Engineering , Agile Methods , Aspect Oriented and Mash-Up," *Int. J. Adv. Comput. Sci.*, vol. 2, no. 4, pp. 20–26, 2011.

[3] N. Amálio and C. Glodt, "A tool for visual and formal modelling of software designs," *Sci. Comput. Program.*, vol. 98, no. P1, pp. 52–79, 2015.

[4] I. Kaur, P. S. Sandhu, H. Singh, and V. Saini, "Analytical Study of Component Based Software Engineering," *Eng. Technol.*, vol. 38, no. February, pp. 441–447, 2009.

[5] Y. Wang, "A process management tool supporting component-based process development and hierarchical management mechanism," *Fifth Int. Conf. Comput. Inf. Technol.*, pp. 906–910, 2005.

[6] A. I. Khan, N.-U. -Qayyum, and U. A. Khan, "An Improved Model for Component Based Software Development," *Softw. Eng.*, vol. 2, no. 4, pp. 138–146, 2012.

[7] T. N. Nguyen, "Component-based software update process in collaborative software development," *Neonatal, Paediatr. Child Heal. Nurs.*, pp. 437–444, 2008.

[8] A. Mokni, C. Urtado, S. Vauttier, M. Huchard, and H. Y. Zhang, "A formal approach for managing component-based architecture evolution," *Sci. Comput. Program.*, vol. 127, pp. 24–49, 2016.

[9] I. Crnkovic, S. Larsson, and M. Chaudron, "Component-based Development Process and Component Lifecycle Technical University Eindhoven , Eindhoven , The Netherlands models process," *Building*, pp. 1–21.

[10] A. I. Khan and U. A. Khan, "An Improved Model for Component Based Software Development," vol. 2, no. 4. pp. 138–146, 2012.

[11] J. Davies, D. Milward, C.-W. Wang, and J. Welch, "Formal model-driven engineering of critical information systems," *Sci. Comput. Program.*, vol. 103, pp. 88–113, 2015.

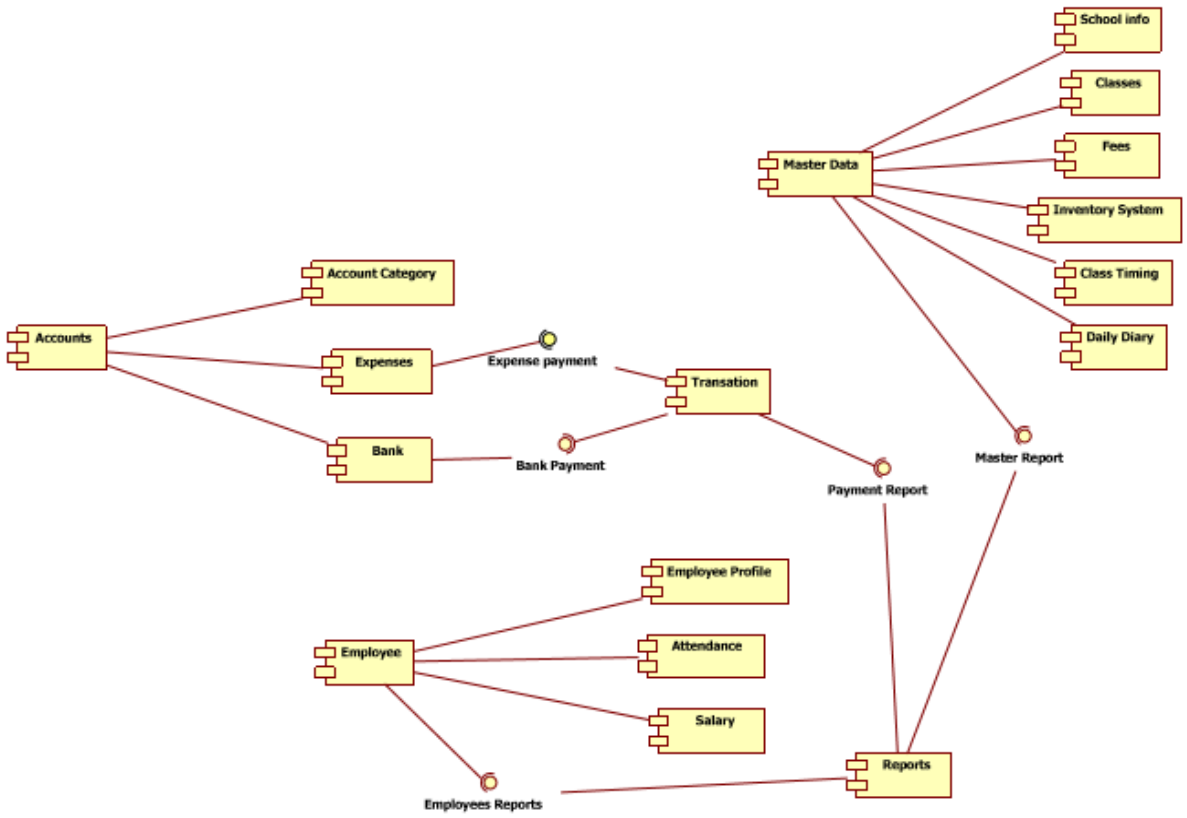


Figure 9: Components model in UML 2.0 School Management System(SMS)