

## Design of Inventory Management System for Herat University Using Component-Based Methodology

Somaia Zabihi

Master of Science, Informatics, Technical University of Berlin  
Herat University, Afghanistan  
E-mail:zabihisomayeh@yahoo.com

### Abstract

The focus of CBSD is to assemble the previously-existing software components and to develop new and large software systems with new characteristics. It emphasizes on the use of Off the Shelf (OTS) components instead of building a system from the scratch. The first objective of this thesis is to discuss about the Component-Based Software Development (CBSD) as one of the advanced methodologies with the example of an inventory management system for Herat University, in Afghanistan. The second objective examines more closely whether CBSD can be a proper approach in Afghanistan, particularly in context of Herat University Inventory Management System (HU-IMS). This paper only focuses on selected phases such as requirement analysis and system design of HU-IMS life cycle. JavaBeans component model and user requirement analysis formulate to design the two separate components of HU-IMS (User Management and Resource Management). Both components are designed in a general way as they either can join as a part of a pre-functional system such as Higher Education Management Information System (HEMIS) or to be used for the purpose to develop new software systems. The results of the thesis confirm that considering CBSD when designing HU-IMS can be a proper approach in the Afghanistan scenario. However, the CBSD, particularly in Afghanistan context, has its own challenges and risks as all other approaches have. Since there are not many localized libraries and repositories available, the first challenge is to find and select the proper component which can be difficult and time-consuming; and the second is the adaptation and localization of the found component insomuch as most of the software developers are unfamiliar with the emerging concept.

**Keywords:** Component-Based Software Development, Herat University, JavaBeans Component Model.

### 1. Introduction

Assembling the previously-existing software components and develop a new and large software with new characteristics is the focus of CBSD (Tang et al., 2011, Cechich et al., 2003). CBSD embodies the “buy, do not build” philosophy espoused by Fred Brooks (Brooks, 1987).

This approach emphasize on the use of Commercial off the Shelf (COTS) components or any kind of Open Source component instead of building a system from the scratch. On the one hand when developers use from existing components, development cost and time to market are be reduced and on the other hand maintainability, reliability of software are be improved (Cai et al., 2000).

Productivity optimization, time to market reduction, quality and usability improvements and complexity management are some advantages of the component based software development approach. However, as other software engineering concepts, it has its own risks and challenges which imperil the success of CBSD projects. Component-Based Software Engineering (CBSE) as systematic approach to component-based development is needed to avoid all these risks and to master those challenges (Crnkovic & Larsson, 2002, Crnkovic, 2001).

### 2. Methods & Materials

The objective of this paper is to find out the answer of research questions: whether CBSD can be a proper approach in Afghanistan and in particular in Herat University Inventory Management System (HU-IMS); and to find out if CBSD is cost and time-effective in all aspects. In order to answer these questions the disciplines of CBSE is applied.

Software development process is comprised of five stages namely as; Requirements Specification, Analysis & Design, Implementation, Test, Release and Maintenance. In the following subsections, requirement and design phases of HU-IMS will be discussed. These phases are selected for the simple reason that the design of a reusable component can be described best through these phases.

### 3. Requirements Analysis and Specification of HU-IMS

#### 3.1 Requirements Analysis

In a component-based approach, it is necessary to analyze whether the available components are appropriate for these requirements. If the appropriate components could not be found, the new components have to be implemented. Whenever requirement engineers are aware of the components that can possibly be used, they could take this risk and negotiate the requirements and recover them in order to use the existing components (Crnkovic et al., 2006).

The case study's target is the inventory office of Herat University where all necessary requirements for developing HU-IMS have been gathered. In order to extract the requirements, several meetings with stakeholders have taken place.

The inventory office seeks an effective and efficient system to aid with managing the numerous devices, stationery, equipment and articles inside the stock. The objective of HU-IMS is to record all the activities related to send and receive of all articles in the stock and report these activities in a certain time (Zabihi, 2013).

During requirements elicitation, scenario is used as a technique to express the particular instances of each important function and behavior of HU-IMS. Therefore in this case, a scenario is brief description of some anticipated or desired use of this system:

#### Scenario

1. Each user has an account for authentication and permission purposes to login to the system.
2. The account enables the user to login to system, registers new item and do some manipulations (Admin user also can create or edit the user).
3. New inventory item can be added by the user and admin user.
4. If necessary, an item can be searched and then can be deleted, edited or deliver.
5. Detailed reports can be provided by the user and admin user:
  - At the end of each season and year, user has to report the asset of the warehouse and the status of inventory item (Report No.F.S.10).
  - Report the surplus items (Report No.F.S.8).
  - Report all the items that have been aided by the organizations (Report NO.M.7).
6. Faculty request their inventory item needs through the F.S.9 form.
7. All system activities shall be saved on log files.

#### 3.2 Requirements Specification

Requirements specification is a general definition of properties which covers the global functional and non-functional aspects which is performed by the developed software system (Madigan, 2006).

##### 3.2.1 Functional Requirements of HU-IMS

According to the scenario and requirement analysis, the system must include these functional requirements:

#### 1. Search

In order to delete, edit and record the process of delivering an inventory item to relevant faculty, the item must be searchable.

#### 2. Manage the inventory item

This functional requirement covers register, edit, delete an item and record the process of delivering an inventory item to related faculty.

#### 3. Provide log file

#### 4. Provide report

#### 5. Faculty requirements response

#### 6. Create or delete user

The main functionality of HU-IMS and the interaction between users and system are described by the UML use case diagram in figure 1 whereas figure 2 depicts the wider functionality of this system. In figure 4 the use cases inside the circle are the focus of this master thesis. These two diagrams provide a specific and complete functional as well as technical view of the HU-IMS.

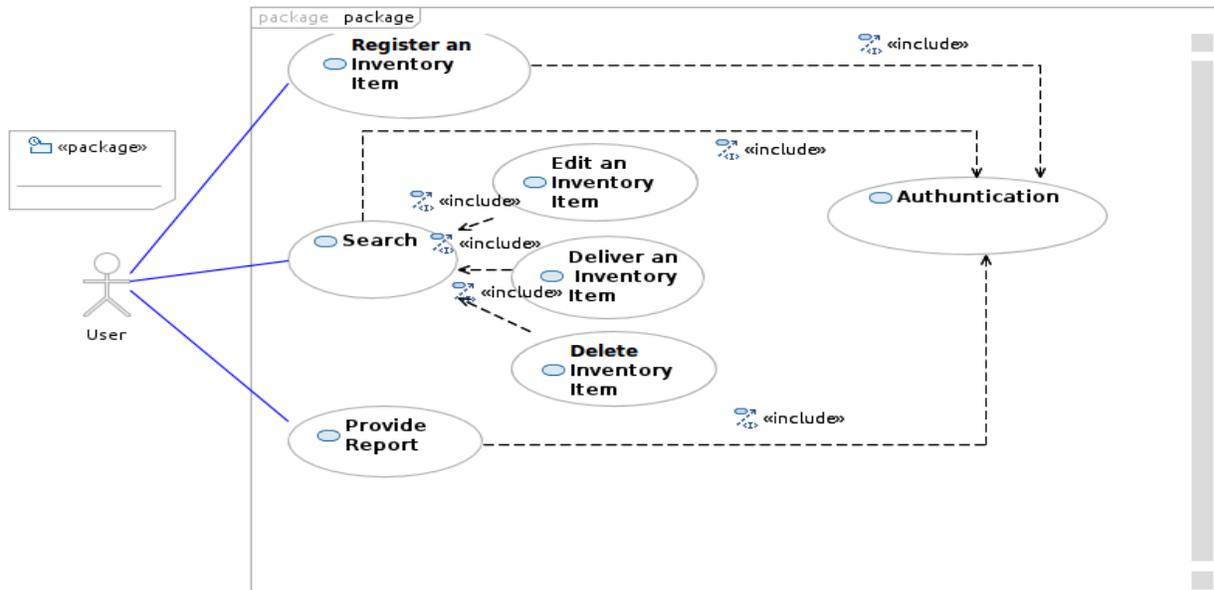


Figure 1. Use case diagram of HU-IMS with main functional requirements

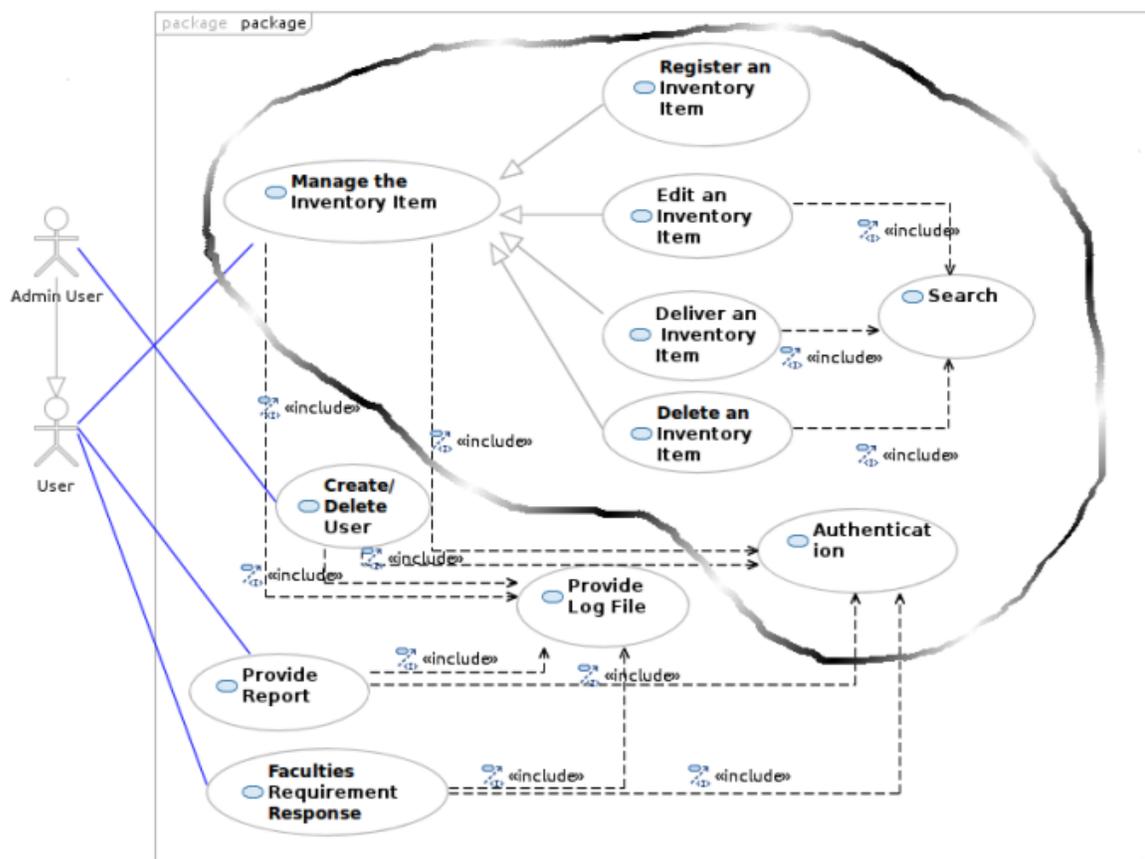


Figure 2. Use case diagram of HU-IMS in more functional requirements

Figure 3 depicts domain model of HU-IMS which defines real world entities in the form of classes and presents the result of problem domain. Figure 3 further identifies Manager UI as boundary; User Controller and Resource Controller as controller; User, Inv. item and Faculty as entity and Order as association class where based on them system is divided into two components; namely as: User Management (User Mgt) and Resource Management (Resource Mgt).

Accordingly, User Mgt component includes User Controller, User and Faculty classes; Resource Mgt component includes Resource Controller, Inv. Item, Faculty and Order classes. The interface of these components, their conceptual implementation and possible connection between them will be discussed with more details in the next section.

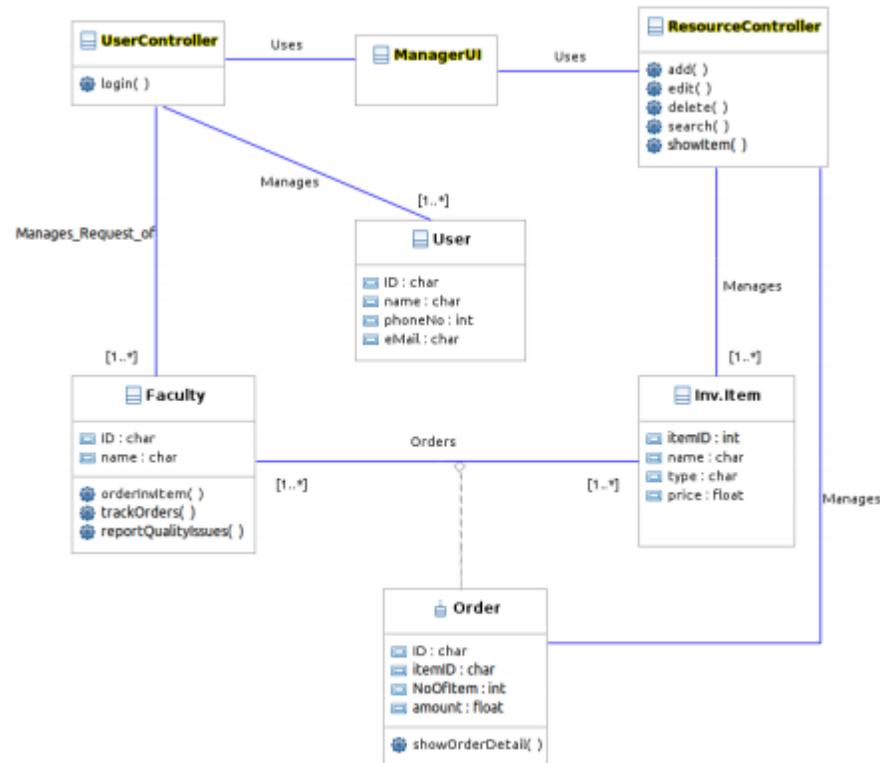


Figure 3. Domain class diagram of HU-IMS

## 4. Result

### 4.1 Component Design of HU-IMS

The goal of system and software design is to establish a technical solution according to the outcomes of previous phases, in order to answer functional requirements of a respective system. At this point, analyzed user's requirements must be translated into technical specifications which depict the design of the system in order to use it as an input for next phases (Pataki et al., 2003).

A reusable component in contrast to a specific proposed component must be designed in a more general way in order to adapt with a variety of requirements of different software. This property leads components to complexity and size increments. In addition, the components must be able to serve particular requirements in an effective way. The combination of these two properties need more design and development efforts, since three to four times more resources are required to develop a reusable component in contrast with the development of a specific purposed component (Crnkovic et al., 2006).

A component model must be chosen to impose the standards and conventions on the task of software architects as well as software development. These models define component types, interaction schemes between components and processes of resource binding.

Accordingly, the JavaBeans Component Model is selected for this case study for the reasons that; firstly, because of its simplicity in contrast to other component models; secondly, its appropriateness for small scale software like HU-IMS and thirdly because of the existence of a well documentation (Estublier & Favre, 2002).

Based on the UML class diagram of HU-IMS (figure 3) and regarding the JavaBeans component model; there are two components namely as: User Management, Resource Management. In the following parts, the interfaces, implementation and assembly of respective components with regard to JavaBeans component model will be discussed.

### 4.2 Interface of a Component

A set of ports provide an external view of a Bean component (figure 4). It is to be noted that, components can communicate with each other through a specific port while these communications are described by the interfaces that the port is supporting. Four types of ports defined in JavaBeans: properties, methods, event sources, and event sinks.

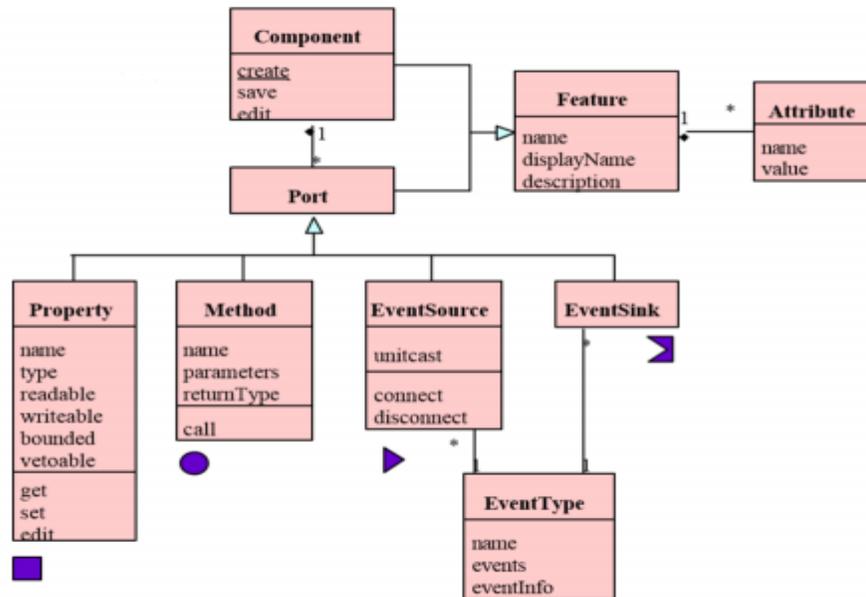


Figure 4. Meta model of the JavaBeans component, (Favre, 2002).

Properties of a bean (JavaBeans component) are basically its associated attributes which can be read or written by calling proper related methods. The notion of the method interface is the same as the concept of normal java method and it can be called from another component (Estublier & Favre, 2002).

Event-based features are a way for component communications. An event is generated in multicast and unicast forms by event sources and is received by event sinks (Estublier & Favre, 2002, Microsystems, 1997).

Figure 4 depicts a Meta model of JavaBeans component with its four types features (Favre, 2002).

Based on the JavaBeans component and interface description mentioned above and with regard to (figure 3) domain model of HU-IMS; User Mgt and Resource Mgt components are designed according to figure 5 and figure 6. In this design, User Management component (figure 5) contains three property interfaces to set and get values to attributes, six method interfaces and three event source ports. Consequently Resource Management component (figure 6) includes three properties, two methods, two event sources and two event sinks interfaces.

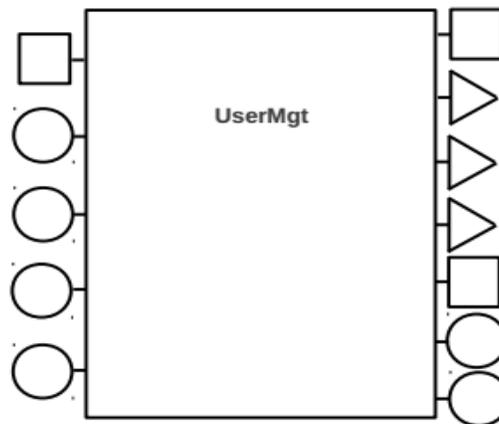


Figure 5. External view of the User Management component with the property

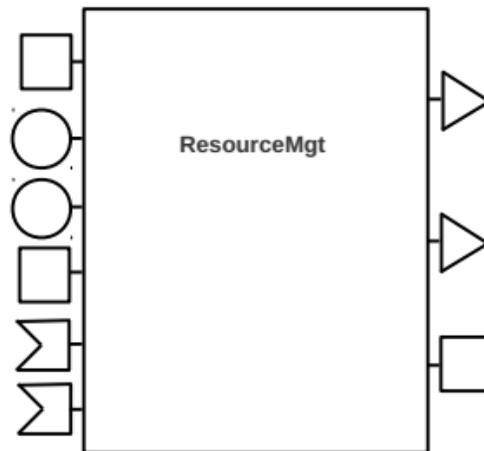


Figure 6. External view of the Resource Management component with the property, method, event source and event sink ports

#### 4.3 Implementation of a Component

The implementation of the component in this section is raised from the conceptual point of view. The implementation of most of Bean components begin with the instantiation of a simple java object and its encapsulation in the components is called single object implementation. In a Multiple object implementation more than one object are encapsulated and collaborated in order to realize a component. According to the Java Bean internal design, *User Mgt* (figure 7) contains two objects; User and Organization and several methods associate to each of them.

As mentioned before, User Mgt formed of User controller, User and Faculty classes; since reusability is the main goal of CBSD and in order to have general design, instead of Faculty, Organization is named as object.

As depicted in the figure 7, each port or interface maps to the methods of these two objects and there are several scenarios. For instance, three source event interfaces could be considered as user Added-event, user Removed-event and create Order-event; three method interfaces as add Organization, remove Organization and order Inv. Item method interfaces and property interfaces are used to get and set attributes to User and Organization objects.

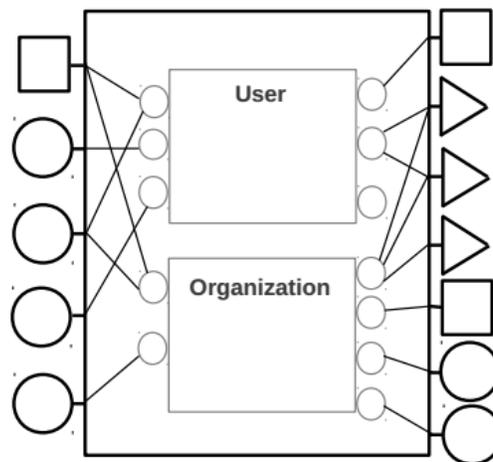


Figure 7. Internal view of the User Management component with multiple object implementations

Consequently, Resource Mgt (figure 8) contains two objects; Item and Order and several methods are attached to each of them. For example, two source event interfaces could be considered as order Generate-event, order Cancel-event and two events sink ports as response Order and cancel Order. Also the method interfaces could present Resource Controller class's methods and Order class's method like add, edit, delete, show Inv. Item and show Order Detail.

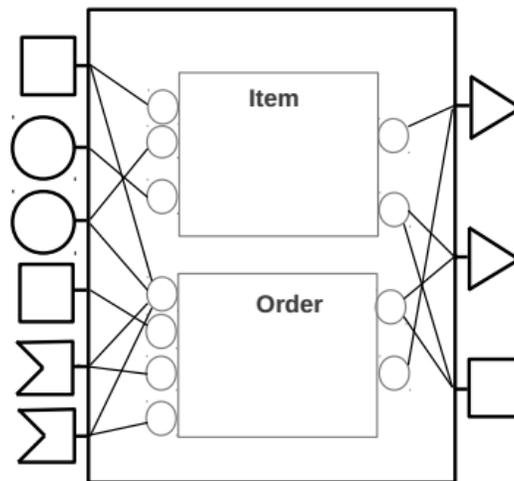


Figure 8. Internal view of the Resource Management component with multiple

**4.4 Component Assembly**

JavaBeans component model does not specify a solution for component assembly, but it supports different ways to assemble Beans (Microsystems, 1997):

1. Programming (component-programming)
2. Scripting language (e.g. Bean Shell)
3. Interactive assembly tool (e.g. Symantec, Visual Age, Net Beans, etc.)

Figure 9 depicts available connections between two components; User and Resource management. There are several scenarios for each case, for example there are two connections between create Order source event of User Mgt and response Order and cancel Order of Resource Mgt. Another available connection could be assembly between order Inv. Item method port of Usr Mgt and get-item (item object) and get-order (order object) property interfaces of Resource Mgt.

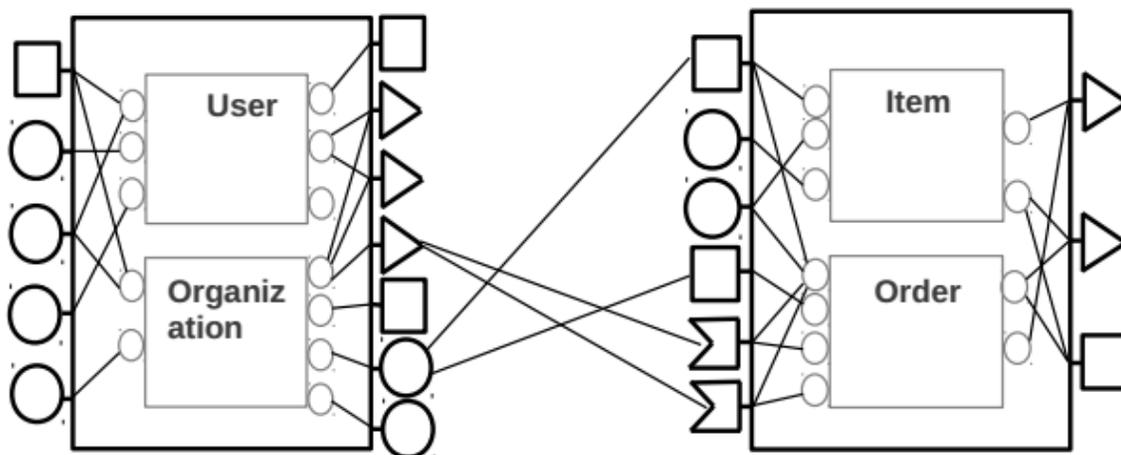


Figure 9. Available connections between two components of UH-IMS and User.

**5. Discussion**

The target of case study was the inventory office of Herat University where all necessary requirements to develop HU-IMS have been gathered. During requirements elicitation, scenario was used as a technique to describe some anticipated or desired use of HU-IMS. Scenario and requirement analysis were used to find out the functional requirement.

The main functionality of HU-IMS and the interaction between users and system are described by the UML use case diagram which provides a specific and complete functional as well as technical view of the HU-IMS; and the work flow – series of actions – of each use case of HU-IMS's UML use case diagram is described by a UML activity diagram.

The domain model of HU-IMS defines real world entities in the form of classes and presents the result of the problem domain. This domain model or class diagram of HU-IMS is the result of problem domain that divides the system into two components; User Management and Resource Management. The next step was to work on solution domain which formed the result of this paper.

JavaBeans component model was selected for the simple reason that there exist a well documentation and simplicity in this model and it is very relevant to the small scale software like HU-IMS.

Now, the most important point is that this component should be reusable. In the context of the inventory management system for Herat University, two components namely User Management and Resource Management are designed in a very general way where they can be reusable for any other Inventory management system at the national and international levels.

The designed components not only can be used for other inventory management system, they can be used for other software system development as well. For instance, User Management component can be used while development of software system for library management.

The other characteristic of this design is that a system is formed as a result of interaction between the User Mgt and Resource Mgt components. This designed system can become part of the other software systems; for example presently there is HEIMS in the ministry of higher education which is comprised of all the information related to the university at the national level and the ministry of higher education itself. Hence, the system can belong to the inventory management part and integrate into HEIMS.

Above mentioned stages confirm, further researches are required firstly in case of the implementation of the Resource Mgt and Adaptation of User Mgt; secondly, the assembly or integration of the Resource Mgt and User Mgt components; and thirdly to test of the system in real world environment which will be university of Herat province in Afghanistan.

## 6. Conclusion

The research questions for this thesis were set to find out about whether CBSD can be a proper approach in Afghanistan specifically in HU-IMS? and also to further find out if CBSD is cost and time effective in all aspects: The facts that these components are designed in most possible general way where they can be used as part of the system like HEMIS or can be used for a new software development purpose proves an increase in the productivity. This in itself can prove to be cost and time effective; because currently Afghanistan is moving ahead towards the computerization, in which case we are in urgent need for various software in different fields like Health, education, industry and etc., meanwhile time there are very limited number of software developers.

Hence, it would be most beneficial to reuse software for to develop other new software. From this aspect it is obvious that CBSD can be a proper approach in the context of Afghanistan specifically in university of Herat in Afghanistan. However, generally CBSD has its own risks and challenges like other approaches and in the context of Afghanistan in particular. Since there are no localized libraries and repositories existing, finding and selecting the proper component is difficult and time-consuming in one hand; and selected component needs more effort in order to get adapted and localized in contrast developing the software from the scratch. Meanwhile most of the software developers in Afghanistan are not familiar with this emerging concept.

## References

- FP Jr, B. (1987). No silver bullet essence and accidents of software engineering. *Computer*, (4), 10-19.
- Cai, X., Lyu, M. R., Wong, K. F., & Ko, R. (2000, December). Component-based software engineering: technologies, development frameworks, and quality assurance schemes. In *Proceedings Seventh Asia-Pacific Software Engineering Conference. APSEC 2000* (pp. 372-379). IEEE.
- Cechich, A., Piattini, M., & Vallecillo, A. (Eds.). (2003). *Component-based software quality: methods and techniques* (Vol. 2693). Springer.
- Crnkovic, I. (2001). Component-based software engineering—new challenges in software development. *Software Focus*, 2(4), 127-133.
- Crnkovic, I., & Larsson, M. P. H. (2002). *Building reliable component-based software systems*. Artech House.
- Crnkovic, I., Chaudron, M., & Larsson, S. (2006). Component-based development process and component lifecycle. In *Software Engineering Advances, International Conference on*, pages 44–44. IEEE
- Estublier, J., & Favre, J. M. (2002). Component models and technology. *Building reliable component-based software systems*, 1, 57-86.



- Favre, J. (2002). *JavaBeans*. Retrieved from <http://www.megaplanet.org/jean-marie-favre/ENSEIGNEMENT/DEA-03-04/AC/JavaBeans-I-scr.pdf>. Visited July 2013.
- Madigan, M. (2006). *Requirements analysis & requirements specification*. Retrieved from <http://ecee.colorado.edu/swengctf/standalone/presentations/ReqsAnalysisWk3.ppt>. Visited July 2013.
- Microsystems. (1997). *JavaBeans*. Retrieved from <http://download.oracle.com/otn-pub/jcp/7224-javabeans-1.01-fr-spec-oth-JSpec/beans.101.pdf>. Visited July 2013.
- Pataki, G.E. Dillon, J.T & Michael, M. (2003). *The New York State Project Management Guidebook*, Release 2. New York State Office for Technology.
- Tang, J. F., Mu, L. F., Kwong, C. K., & Luo, X. G. (2011). An optimization model for software component selection under multiple applications development. *European Journal of Operational Research*, 212(2), 301-311.
- Zabihi, S. (2013). *Component-Based Software Development Exemplified by an Inventory Management System for Herat University, Afghanistan*. 3659578665, LAP Lambert Academic Publishing.

### Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal. This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).

