A SOFTWARE REQUIREMENT ENGINEERING TECHNIQUE USING OOADA-RE AND CSC FOR IOT BASED HEALTHCARE APPLICATIONS

Aviral Srivastava, Fenil Patel and Prof. M Sivagami
Vellore Institute of Technology, Chennai campus,India

ABSTRACT

This Internet of things is one of the most trending technology with wide range of applications. Here we are going to focus on Medical and Healthcare applications of IOT. Generally such IOT applications are very complex comprising of many different modules. Thus a lot of care has to be taken during the requirement engineering of IOT applications. Requirement Engineering is a process of structuring all the requirements of the users. This is the base phase of software development which greatly affects the rest of the phases. Thus our best should be given in the engineering of requirements because if the effort goes down here, it will greatly affect the quality of the end product. In this study we have presented an approach to improve the requirements engineering phase of IOT applications development by using Object Oriented Analysis and Design Approach(OOADA) along with Constraints Story Card(CSC) templates.

KEYWORDS

Constraint Story Cards, OOADA-RE, Medical and Healthcare in IoT, Requirements Engineering

1. INTRODUCTION

This Internet of Things suggest only one phrase: Any service anywhere! In IoT, the information is shared and gathered among various devices and cloud, making it possible to store and analyze the data for various applications. IoT in Health care systems shows a lot of promise in improve the quality and efficiency of the care provided. The patient data will be collected through multiple sensors which will be recorded and analyzed using microprocessors and controllers.Fig 1 [1] illustrates the use of IoT in Medical and Healthcare. The increased connectivity of the system through cloud computing gives doctors and caregivers the ability to access realt time information and assist them to make informed decisions. The diseases can be managed efficiently and can be prevented from going out of hand through continuous patient monitoring and providing the caregivers with real time information. Building such a complex system demands a great amount of attention. The first step in developing the system will be the requirements engineering. Requirements engineering is a step in software development cycle where all the requirements of the user and system are structured. During this step the goals and functionalities of the end product are described. The designers and developers use these specifications as the road map. Thus it is very important that the requirements defined are clear, easy to understand and consistent. OOADA is a requirement engineering activity used to create a model of functional requirements of the system. In OOADA, first step is to find and organize the objects then later on the actions are identified defined for the objects. Section 2 deals with Related Work, section 3 with the proposed RE technique and section 4 applies the proposed technique in Medical IoT applications.

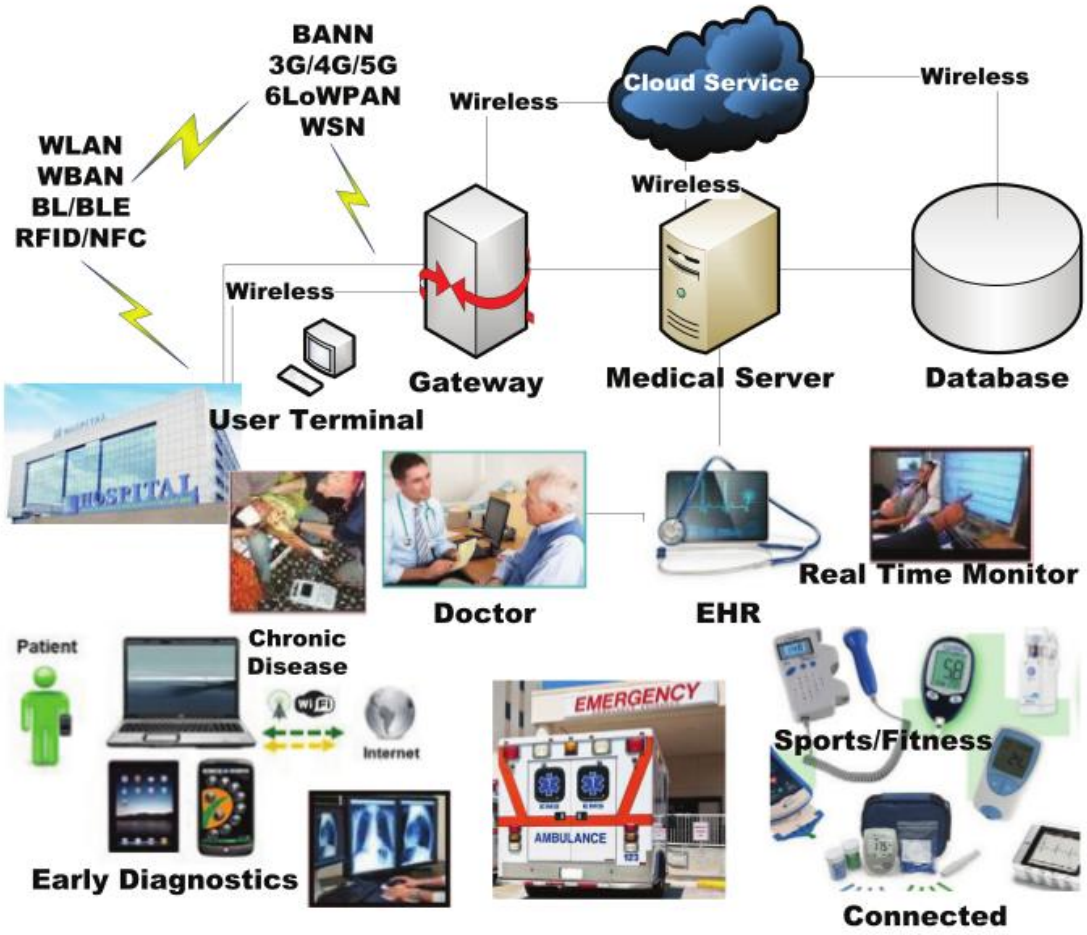

FIGURE 1. Healthcare trends.

Figure 1. IoT in Medical and Healthcare

2. RELATED WORK

The Requirements engineering is the first step in most of the software development cycles. J.M.A. Shari [2] et al have actually concluded that the requirement engineering step has the maximum effect on the final product and the poor RE leads to end product with poor quality. H. Nakajo et al [3] proposed an OOADA-RE(Object Oriented Design and Analysis Requirements Engineering) Software Engineering technique stating this as a process in which different processes and techniques are applied to identify and structure the requirements of the desired system. As explained earlier it is very important to specify clear, easy to understand and consistent requirements during this phase, because it is more expensive to fix the errors and modify the system requirements in the later phases of the software development cycle.

There are many approaches for specifying requirements and transforming them into UML diagrams.[4] Amit Raj, et al presented a approach where they transformed the requirements specified in Object Management Group's standard Semantics of Business Vocabulary and Rules (SBVR) framework into UML models. M.G Illieva et al .[5] introduced another approach where the used natural language processing to remove ambiguity, contradiction in the requirements specifications and to map them to UML diagrams.

Another popular approach for requirements specification was scenario based requirements engineering. The author N.M.Ian F. Alexander [6] introduced the Software Requirements Engineering technique which defines the requirements in terms of scenarios. The problem with this approach was that it was difficult to map the scenarios into uml diagrams. The OOADA-RE approach was introduced by A.Zeaaraoui et al [7] in order to fill this gap. Using this approach it was more convinient to build UML diagrams.

The UML diagrams from this approach were lacking associations. Walid Dahhane et al [8] introduced Constraint Story Cards(CSC) template approach for creating associations between the objects. Discussing about the work done in requirements engineering of IoT, there has not been much work done which uses the concepts of OOADA and CSC and in this paper we have presented the same.

3. PROPOSED WORK

A new software Requirements Engineering technique has been introduced for IoT based health-care applications using OOADA-RE and Constraint Story Cards(CSC). In this section we have demonstrated the step by step procedure to use OOADA-RE and CSC for requirements engineering of minimal Health-care IoT application.

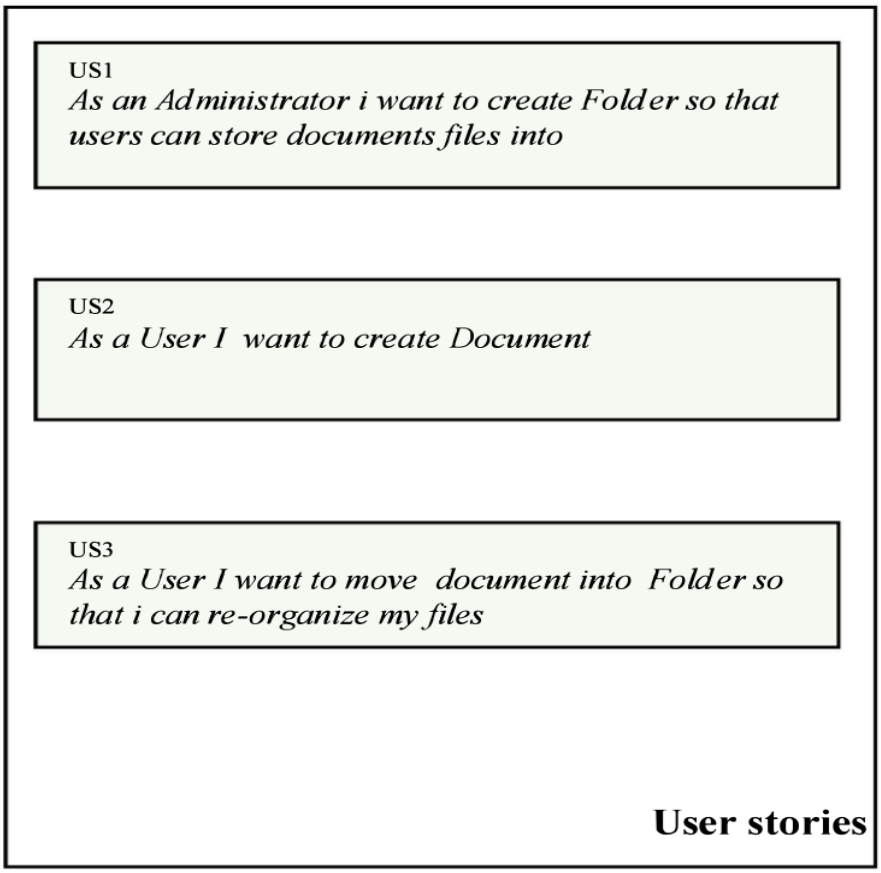

Figure 2. User stories

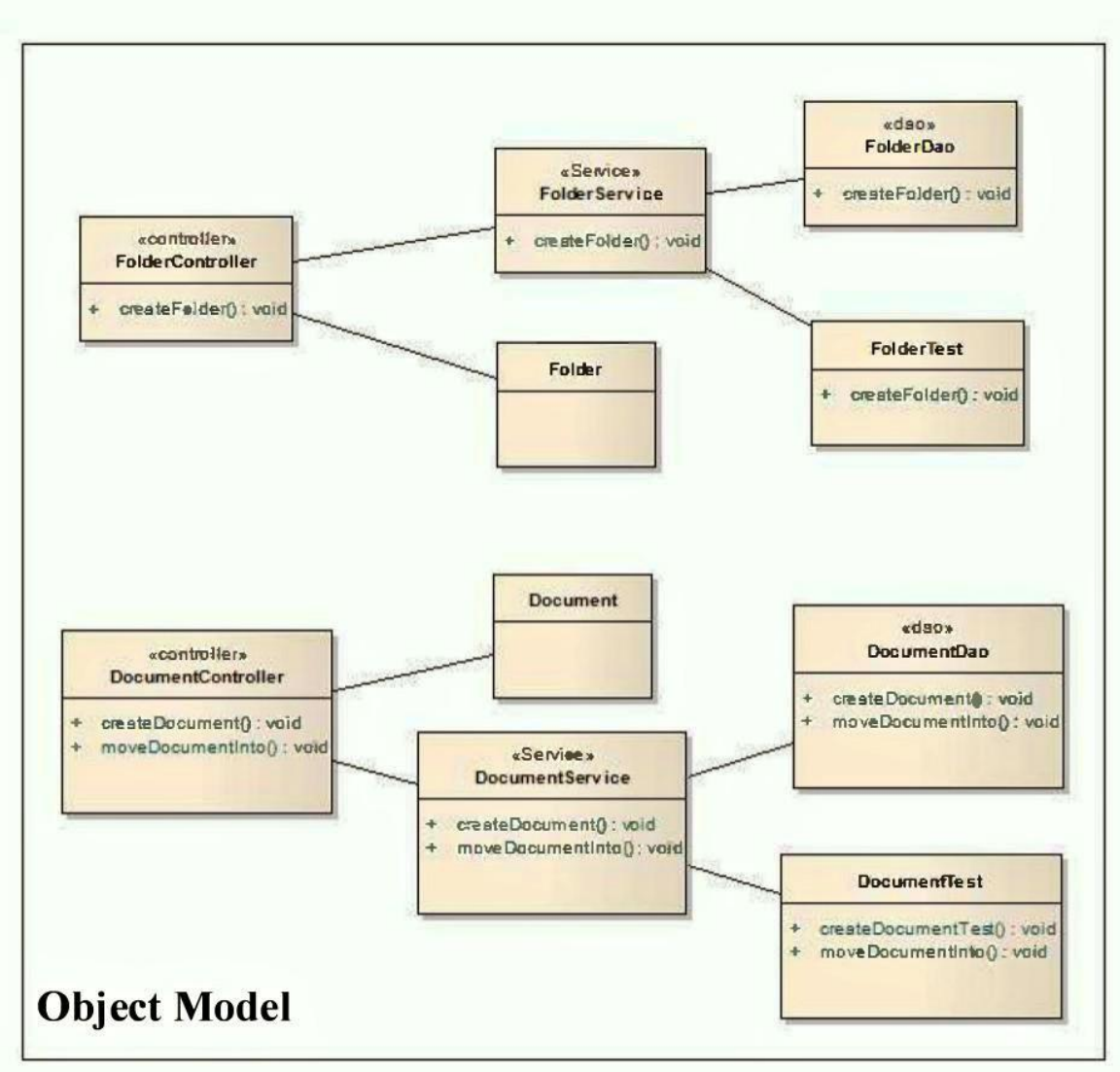

Figure 3. Generated Object Model from User Stories in Fig.2

3.1. OOADA-RE

In OOADA-RE we use user story templates such as: ”As a <role>, I want to <action> <object>, so that <business values>”. [7]

This templates can be used to map the requirements to the class diagrams. Using the story templates defined by the users in the figure 2 [7] using the above mentioned template, we can generate the class diagrams as illustrated in the figure 3 [7]. We first identify the objects from the user defined stories and organize it. Then in the next step we identify all the actions defined for the objects and map the objects and actions onto the diagram.

3.1.1. Advantages and limitations of OOADA-RE in contrast with other techniques

As discussed in the Section 2, many various types of techniques have been proposed for effectively specifying requirements and transforming them to UML diagrams. The major drawback of scenario based approaches is the non-possibility to map scenarios from this approaches into object model diagrams. Generally scenarios describes functional nature of the

system and cannot be effectively mapped to object models. On the other hand, OOADA-RE approach makes it possible to map the requirements into object models by using the templates which can help in identifying classes, attributes etc.

Long paragraph based approaches for defining requirements and use cases many times lead to ambiguity which in turn affects the quality as well as build time of systems. On the other hand, OOADA-RE with the help of minimal user story templates minimize such ambiguities in the requirements.

Dependency among the classes is one of the most important part of any object class diagram. The major drawback of the OOADA-RE method is that it does not deal with the dependencies. The Constraint Story Card templates described in the subsequent section deals with this drawback.

3.2. Constraints Story Card Templates

The class diagrams generated by OOADA-RE lack associations between the classes. Associations are very important part of any class diagram. They defined how all the objects interact with each other. Constraint Story Cards(CSC) templates are used for filling this gap. In CSC there are four templates for 4 types of associations[8]:

• Simple Association template
• Inheritance template
• Aggregation template
• Composition template

The four templates uses a generic formula which has following arguments:

• Source Object - Source object of the association.
• Role - Association role between the objects.
• Association Expression - for example ’has’, ’is a’, ’is composed of’.
• Cardinality - Cardinality can be expressed by numbers or words like ’one’, ’many’.
• Target Object - Target Object of the association.

4. CASE STUDY

We demonstrated the proposed Software Requirements Engineering Technology. Any IoT application requires these classes and more classes can be added (as per the requirements) later: (Doctors, Patients, readingsReport, reportReview, Admin). readingReports class holds the reports generated from the sensor readings. reportReview class holds the reviews given by doctors on the generated readingReports.

• Step 1: Forming User Stories

In this step we will gather all the functional requirements of our IOT application through meetings, workshops, interviews etc. The we will format the gathered information as user stories. Here we will use a part of user stories for creating a Medical IOT application. The box in figure 4 contains some user stories for the application.

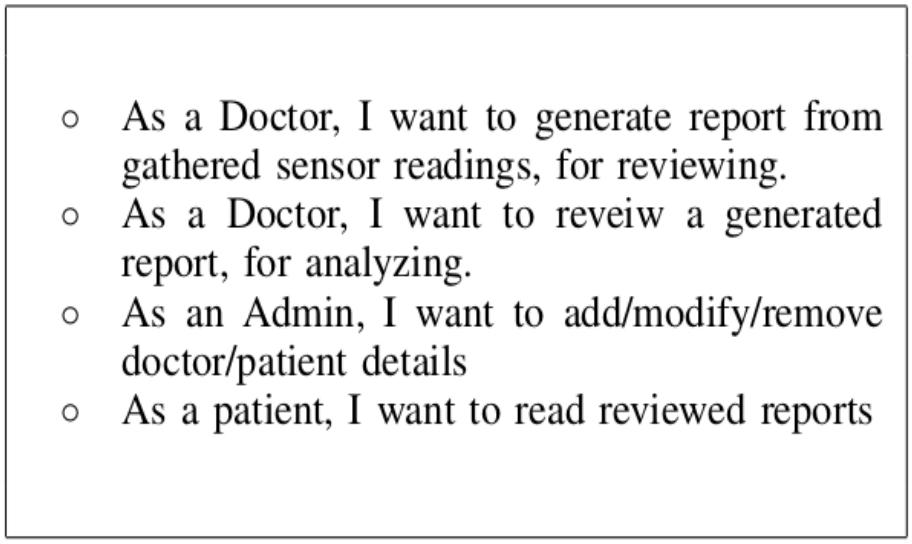

Figure 4. Sample user stories

Step 2: OOADA-RE

We start by categorizing the entities. For example the entities into object, actions or business values. For first user sroty i.e ”As a Doctor, I want to generate report from gathered sensor readings, for reviewing.” Doctor is an object, GenerateReport is an action and ”for reviewing” is the business value. Then we decide what actions must be added to the object to fulfill the user’s requirements. In our case the requirements specify that the Doctors should be able to generate report and review report, therefore we will include this two actions for Doctor object. Similarly we can generate a class diagram from this data by creating their objects and adding actions to them.We can generate the class diagram as illustrated in figure 4. Now we will use CSC templates to create the association between the classes.

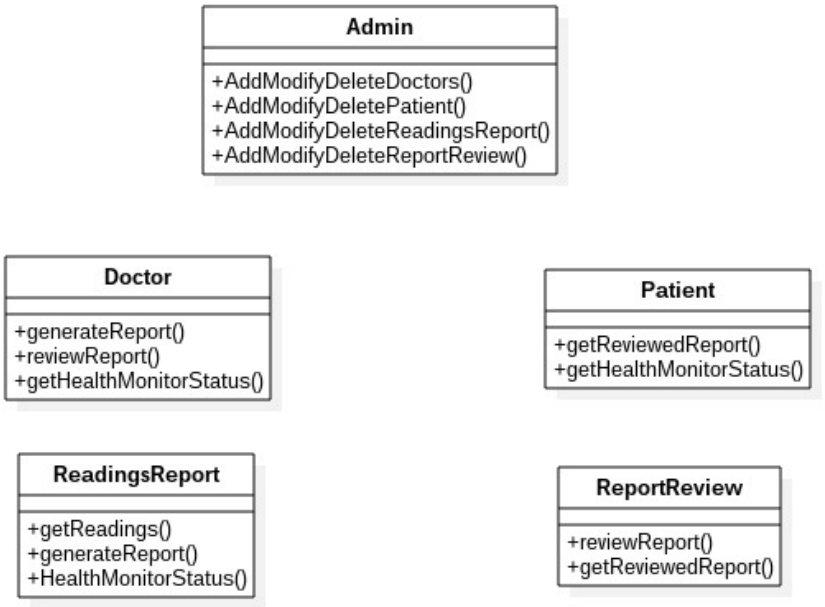

Figure 5. Transformation of user stories to object model using the OOADA-RE approach.

Step 3: Using CSC template for class association As we can see that figure 5 lacks association, which are very crucial. These associations define how the objects interact with each other. It is very important to capture these interactions during the requirement engineering phase as they will be very helpful during the development and implementation phases. As explained earlier there are four CSC templates Simple, Inheritance, Aggregation and Composition.

- Simple Association Template
  This template deals with simple associations. For example, a doctor can be associated with many patients and a patient can be associated with many doctors. We can express such associations using following template:
  [Role] <Source Object> has or verb <cardinality> [Role] <Target Object>
  For example:
  Doctor treats|has one or many **Patient**.

- Inheritance Template
  This deals with the inheritance association i.e parent-child relationship between classes. The inheritance association uses IS-a relationship.
  <Source Object> <is a> <Target Object>.
  For example:
  Doctor is a **Person**.
  Patient is a **Person**.

- Aggregation Template
  Aggregation is a special form of association where objects have life cycle as well as ownership. We use following template for aggregation.
  [Role] <Source Object> contains <cardinality> [Role] <Target Object>.
  For example:
  IOT Application contains many **Sensor Modules.**

- Composition Template
  Composition is a special form of aggregation where if parent object is destroyed, child object cease to exist. It is actually a strong type of aggregation aslo known as death relationship. We use following template for composition.
  [Role] <Source Object> < is composed of > <cardinality> [Role] <Target Object>.
  For example:
  Patient report is composed of one or many **Doctor Reviews.**

We can use this templates to introduce associations in our OOADA-RE generated class diagram. This templates can also be used for big complex applications. For our simple IOT application example, the final class diagram will be as shown in Fig 6.

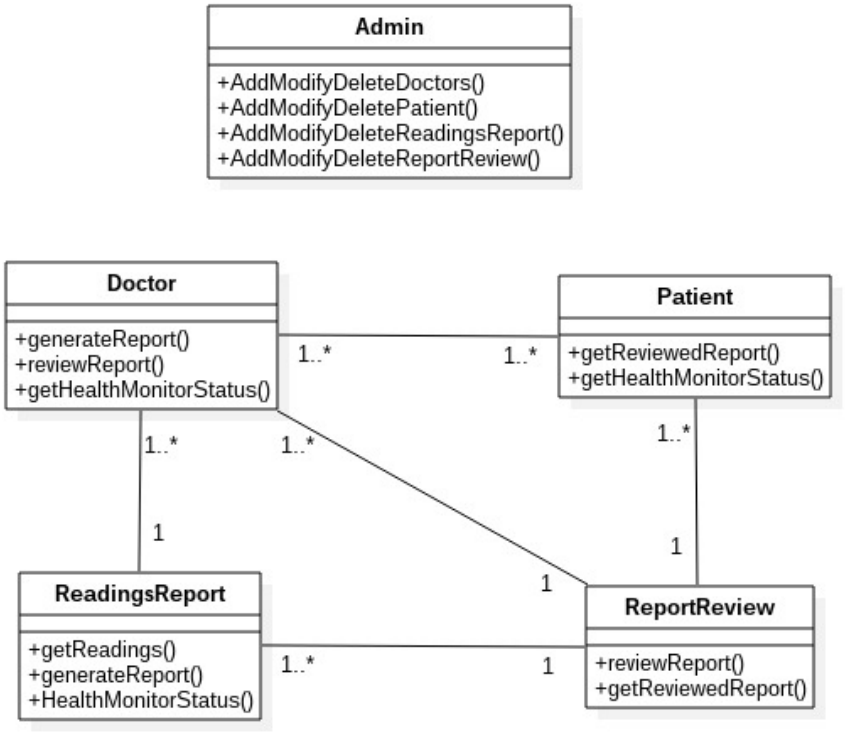

Figure 6. Final Class Diagram using OOADA-RE and CSC.

Step 4: Implementation and generating code: In this step we transit from requirements engineering to implementation phase. But before doing that we must detail out diagram with proper datatypes, attributes etc.

We have seen OOADA-REs usage in Requirements Engineering which helps us developing the structure and model for RE and showed the break they present when moving to construction, we have seen Constraint Story Cards aimed to handle associations. In this paper, we have suggested the combination of OOADA-RE and CSC for IoT (Medical and Healthcare) applications which will follow the roadmap of OOADA but the users shall be given the better templates for user stories; the association between classes can be well drawn. Hence, we propose this method to be working better than the rest two separately or individually.

5. CONCLUSIONS AND FUTURE WORK

For complex application using IOT deserves great amount of attention in requirements engineering. In this paper we introduced a step by step procedure to model the requirements of users to a proper detailed class diagrams using Object Oriented Analysis and Design Approach(OOADA) and Constraint Story Card(CSC) templates. In this paper we concentrated on IOT applications for Medical and Healthcare, in the future we will make this more generic for all IOT applications.

REFERENCES

[1] M. H. K. M. H. K.-S. K. S. M. RIAZUL ISLAM, DAEHAN KWAK (2015), "The internet of things for health care: A comprehensive survey," in IEEE Access (Volume : 3), pp. 678–708, IEEE.

[2] J. M. A. Shari Lawrence Pfleeger (1998), "Software engineering: Theory and practice,".

[3] H. Nakajo, T. Kume (1991), "A case history analysis of software error cause effect relationships," Transactions on Software Engineering, pp. 830–838.

[4] S. H. Amit Raj, T. V. Prabhakar (2008), "Transformation of sbvr business design to uml models," ISEC '08 Proceedings of the 1st India software engineering conference, pp. 29–38.

[5] O. O. M. G. Ilieva (2005), "Automatic transition of natural language software requirements specification into formal presentation," Natural Language Processing and Information Systems. NLDB 2005. Lecture Notes in Computer Science, vol 3513., Springer, Berlin, Heidelberg.

[6] N. M. Ian F. Alexander (2004), "Scenarios,stories, use cases: Through the systems development life-cycle,".

[7] M. B. T. B. A.ZEAARAOUI, Z.BOUGROUN (2012), "Object-oriented analysis and design approach for requirements engineering," in Second International Conference on the Innovative Computing Technology (INTECH 2012), pp. 133–137.

[8] E. H. E. T. B. Walid DAHHANE*, Adil ZEAARAOUI (2014), "An automated object-based approach to transforming requirements to class diagrams," 2014 Second World Conference on Complex Systems (WCCS),pp. 158–163.