EXTRACTING USE CASE DIAGRAM FROM REQUIRMENT ENGINEERING PROCESSES

Lena Khaled
Software Engineering Department, Zarqa University, Amman, Jordon

ABSTRACT
Part of the solution to requirement elicitation problems is to structure it as a diagram. Use-case diagram is a simple tool that represents the system as a black box and describes the communication between a system and its environment. This paper describes how use-cases can be extracted from requirement engineering processes; it illustrates the main phases of the requirement engineering then pull out the concepts that form the use case diagram from these phases.

Keywords: Requirement Engineering, Requirement Engineering Processes, Use Case and Use Case Diagram.

1. INTRODUCTION
Software process is the set of fundamental activities that leads to building software products; these activities may engage the development of software from scratch to a standard programming language such as any language based either on function approach like C language, or based on object oriented approach like Java language [1].
The fundamental activities to all software process models are [1]:

- Software requirement: the functionalities of the software are specified through this step.
- Software design and implementation: design model of the specification must be constructed through this stage.
- Software validation: validation step must be done to ensure that the product does what the customer want
- Software evolution: the software product must evolve to meet the customer's change.

Clearly, the requirement activity is the first stage and all other activities done after, so it must be done very carefully because all other activities are based on it.
This paper is constraint on capturing requirements of the software product through use case because it takes the requirements through the user's point of view that, eventually, is the best way to solve the right problem.
This paper is organized in the following way: Section 2 shows an overview of related works on use-case and requirement, section 3 defines the road map of the requirement engineering process, section 4 describes use-case, section 5 illustrates the extraction of use case from requirement phase and finally, the conclusion is presented in section 6.

2. RELATED WORKS
Many different researches work on use-cases. A. Issa [2] constructed a new use-case patterns catalogue. This catalogue has then developed a new framework for requirement engineering. There, users move up a number of concerns that have been considered to outline further phases of this work. [3] This work described that use cases and scenarios are different each other but they play a complementary roles in requirement engineering. It defined use cases as a collection of scenarios, it shows that use cases are appropriate for describing the behavior of the system requirement when interacting with actors while scenarios can be used to validate functional and non-functional requirement specification, this work also presented integration of scenarios and discussed the strength of this type of integration. [4] This work showed that the effective way to express requirement elicitation and analysis is use case. It presented the environment of use case based requirement engineering which is called UCED(Use Case Editor),this type of editor included tools for use case edition in conjunction with domain model, use case integration and use case simulation. [5] This work described use cases and scenarios as an analysis tool during requirement engineering activities, this is because their richness and informality. The description is done through explaining a goal driven analysis of requirement specification for an electronic commerce application, also it defined the specific risks that appeared through description. It also discussed the impact lessons that learned for requirement engineering in the context of building such quality systems during goal and scenario analysis.
This paper lays out the main idea of capturing functional requirements through use cases; it describes the road map of requirement engineering then shows how use-cases are used to define the requirement of the software product.

3. REQUIRMENT ENGINEERING (RE): A ROADMAP

Requirement engineering is the process of discovering the purpose of software by identifying actors of the software and their needs, and then it writes the documentation in an agreeable way to analysis and later to implementation. The actor's need may vary and conflict depending on his view of the environment and the task he wishes to complete. This definition is important to number of users for many reasons. First, it constraints on actors and their needs. Second, the definition refers to specification's evolution over time to solve the conflicts of actors' needs [6].

Therefore, the benefit of good requirement engineering includes the agreement among developer and customers such that the delivered system meets the customer's requirements [7].

3.1 REQUIRMENT ENGINEERING PROCESSES

Good requirement practices make software development faster. Moreover, it protects a project from failure. Software requirements encompass two major phases; these are [7, 8]:

- Requirement definition
- Requirement management

Requirement definition phase is the collaborative process of collecting, documenting and validating a set of requirement that represent an agreement among users, so this phase is divided into sub phases, and those are:

- Eliciting requirement
- Analyzing requirement
- Documenting and validating requirement.

In other side, requirement management phase involves working with a defined set of the requirements, also it includes managing changes to that set of requirements through out the life cycle of the project and this is what we call the evolving of the requirement so this phase include one sub phase that is evolving requirement.

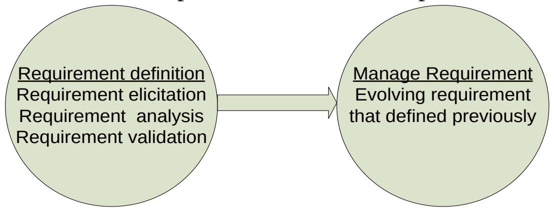

Figure 1. Requirement engineering phases

4. USE CASES

Use-cases were introduced to the software engineering world by Jacobson and his team in 1992; it is one of the nine diagrams that composed UML. Use-cases form the basis of how system interacts with the external environment. It is also a graphical representation between actors and use-cases and between a use-case and another [9]. Use cases are high –level descriptions of system processes and they organized functional requirements in ways meaningful to users and they designed to help ensure that the behaviour of the system is what the users require.

There are three main things needed to describe a use-case [10]

- The actor: it is a type of user that interacts with system.
- The system that is being used which is described as a black box system.
- The functional goal which represents the reason for using the system.

Some essential things must be known:

- Actors must be external to the system itself; they might be other systems not just users.
- The goal must be worth to the actors.

So when analyzing functional requirements for a system, the main questions are:

- Who will be using the system?
- What is the reason for using the system?

4.1. WRITING AN EFFECTIVE USE-CASE

A great way for writing an effective use-case is to do several steps:

Step1: Define actors

Many users interact with any created system. The basic users of the system must be chosen; such as in E-commerce application, the general basic users are buyers and sellers. The visual notations for these actors are based on UML notations and are represented in figure 2.

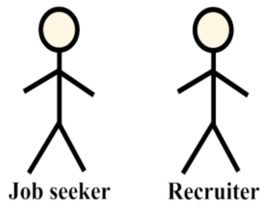

Figure 2. Actors' notations

Step2: Define the actor's goal
After presenting the main actors, the next step would be defining their goals when using the system. In requirement management, refinement always done to the goal. Effective use-cases should have understandable actors and goals. The goals to the above actors are shown in figure 3.

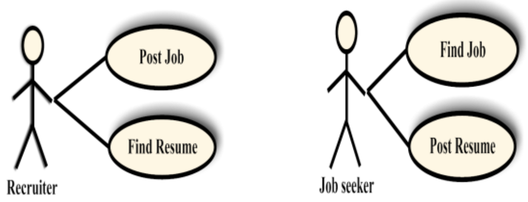

Figure 3. Actors and their goals

Step3: Define a reuse opportunity for use-cases
In this step, the term general actor is defined clearly. For example, a "man" is said to be general, but not as general as a "person" is. In E-commerce system, we see the redundant behaviour for both seller and buyer; both of them must search list and create an account. Instead, we can create a person to make these functions and then both seller and buyer inherit these functions from this person as in figure 4.

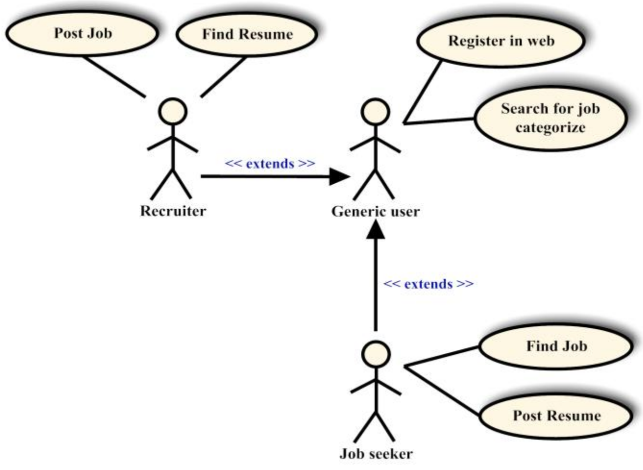

Figure 4. The reuse in use case model

The extend relationship means that the flow of information is only done under specific conditions.

Step4: Name and briefly define use-cases
Usually the name of the use case must be simple and must be in a clear English language.

Step 5: Create the use-case basic flow
Here basic flow represents the most important events occurred through interacting between actors and users.

Step6: Produce use-case document
The purpose of this step is to transfer knowledge from domain to the software developer.

5. RESULTS

A use case is a technique for documenting the requirements of the software system. It describes the general behaviour of the system. Because requirement engineering is the main and first step of building the system then we need to extract the use cases from requirement engineering to structure requirements in a more clear way.
As we said in the previous sections use case built on three dimensions so building these dimensions from requirement engineering (to complete the diagram of use case) is done through the following:

- From requirement elicitation we indicate to the stakeholders (or actors) of the system. Either all actors affected from system directly or indirectly. We can elicit the requirements from those actors that use the system.
- From requirement documentation we can extract all functions that made by actors when interacting with system, this functions is not the internal work of the system but it is only showing the steps that users follow to perform the functions, these functions represent the use cases.
- The main goal is representing by requirement definition, what we document here is the solution requirements that define goal or objectives of any software system.

Table 1 represents the requirement engineering processes also it represents what we can extract to complete the basic terms that needed to build use cases.

Table1. Extracting use case from requirement engineering process

Requirement engineering processes	**Use case terms**
Requirement definition	**Define goal or objective of the system**
Requirement elicitation	**Define stakeholders(actors)**
Requirement documentation	**Define use cases**

6. JOB AGENCY SYSTEM(JAS): AN EXAMPLE

Job Agency System(JAS) is a one type of Business to Consumer (B2C) commerce that transact services from business to individual, it is very effective system because it uses internet regularly ,also it is very fast and easy according to traditional system.

Following steps show how we extract use cases to JAS from requirement engineering processes:

- As a first step of requirement engineering processes is requirement definition; to define the requirement of this system we need to define the purpose of this system. The main objective is serving people who are looking jobs.
- Requirement elicitation is a second part we apply, to elicit information this done only through stakeholders of the system. Although many stakeholders (actors) shared with this type of the system (like advertising agencies, brand mangers, marketing mangers), only two main actors play the basic roles in the system, those are job seekers and recruiter. The main functions of recruiter are posting jobs, finding resumes, and finding the feedback of the interview. While the main functions of the job seekers are: finding job, job upload and resume, online interview and take the feedback of the interview. To document requirement we build a use case diagram for JAS to define all use cases of the actors through it. Figure 4 illustrates use case diagram of JAS.

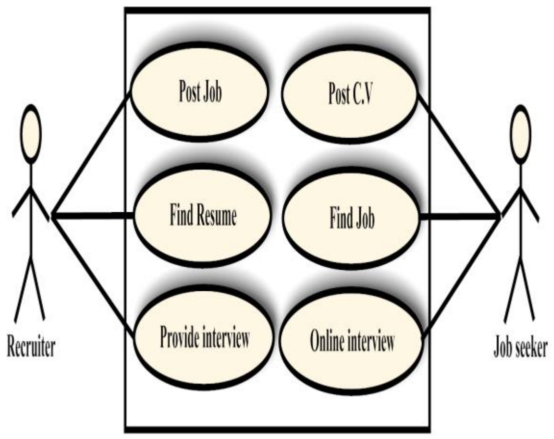

Figure 5. use case diagram for JAS

7. CONCLUSION

The most important step in building any system is capturing its requirements. Developers must elicit the information of the system from users or persons that used system, analyzed the information they elicit then verify all the work. To make requirements clear it must be represented in some structure way but it must stay consist, complete and understandable. Use case is the most appropriate way to express these requirements. This paper described how use cases extract from requirement engineering processes to represent the functional requirement of any system.

REFRENCES

[1]. I. Sommerville, "Software Engineering,"Eighth Edition, Addison-Wesely,(2007).
[2]. A. AL-Ali,"Use Case Patterns Driven Requirment Engineerin," IEEE 2nd International Confernce On Computer Research and Development,(2010). DOI: 10.1109/ICCRD .2010.16.
[3]. S. Some, "Enhancement of Use Cases Based Requirments Engineering approach with Senarios," proceeding of 12th Asia-Pacific Software EngineeringConference9(APSEC'05),IEEE,(2005).
[4]. S. Some, "An Environment for Use Cases based Requirements Engineering," proceeding of 12th IEEE International Requirements Engineering Conference, PP: 364-365, (2004).
[5]. A. Anton,"Manging Requirements During Goal-Driven Requirement Engineering: Challenges Encounterd and LessonsLearned", IEEE International Conference on Software Engineering,ICSE,(2000).
[6]. B. Nusibeh and and S. Easterbrook,"Requirements Engineering: A Roadmap," Proceedings of the Conference on The Future of Software Engineering, USA, (2000).
[7]. M. Dorfman, "Requirements Engineering, "IEEE Computer Society Press,(1997).
[8]. A Borland white paper, "Effective Requirements Definition and Mangment,"(2006).
[9]. D. Kulack and E. Guiney, "Use Cases Requirements in Context," Person education, 2nd edition, (2004).
[10]. J. Gorman, "Use cases –An Introduction,"(2006). www.parlezuml.com.