A GRANULARITY-ORIENTED USE CASE ANALYSIS METHOD

ZHANG NA
Software Engineer Institute, School of Computer, Beihang University

LUO YANJING
Software Engineer Institute, School of Computer, Beihang University

ABSTRACT
The emergence of use case analysis method provides an effective way of capturing and analyzing software requirements, but for beginners, how to use it efficiencily is always a problem. In paper, it presents some commom mistakes that users often make and principles and standards users should comply with. Finally, it proposes a granularity-oriented use case analysis method in detail.
KEY WORDS
Use case analysis method, Use case, Scope, Actor, Granularity of use case

1. INTRODUCTION

Use case analysis is a method of describing system requirments, especially system behaviors [1]. It asks developers, from requirment analysis phase, to consider from the user point of view. It is easy to write and understand use case, but for beginners, how to analyze and decompose use case and grasp its granularity are difficult. Therefore, a feasible method is needed.

2. USE CASE ANALYSIS METHOD

Use case is a series of descriptions of actions including variables, and used by system to create valuable and visible results to users [2]. Essentially, it is one interaction between user and system for certain goal. Set of use cases constructs system requirments, and set of use cases functions constructs system function [3]. Actor is outside system and communicates with it. An actor is a tight set of roles acted by users when they communicate with related use cases. Actor can be human beings, hard devices, or auto-systems. One important source of actor is stakeholder, who directly sends action to system or recives feedback from system. It is important for requirment analysis and system design to accurately determine scope and control scope granularity. In general, scope comes from system goal, such as business goal, system feature.

Actor, scope and use case are interdependent. Once scope is determined, actor can be defined easily. Because use case is determined by actor's entire desire, then use case can be captured quickly.

3. STATUS QUO OF USE CASE ANALYSIS

3.1 COMMON MISTAKES

It is easy to write summary use cases, but hard to study details of them and list steps of every use case scenario. So analysts often make the following mistakes.

(1)Take system step as use case. Using step to decompose use case can cause inaccurate requirments and too small granularity.

(2)Use case is system function. System function is treated from developer's or systemic perspective, and function decomposition is a vertical view. Use case, however, is treated from user's perspective and it has a definite goal, so it is a lateral view [4].

(3)Use case is system requirment. The implied requirments in use case are function requirments, and cannot present safty requirments and so on.

(4)Confuse goal and steps of achieving the goal. One use case is actor's one desire, one entire event for target system. It needs several steps to finish the event, but the steps cannot completely reflect actor's goal.

(5)Use cases in same requirment phase have different granularities. That is because analysts fail to grasp scope and which abstract level is current phase in.

3.2 EXISTING ANALYSIS METHODS

Alistar Cockburn proposes an effective method [5]. Firstly, analysts decide system scope and actors and list users' goals, and then obtain summary use case by adding, subtracting or combining user goals. For every use case, analysts take breadth-first algorithm to extend it from low-precision to high-precision, including getting stakeholder, precondition and guarantee, writing main success scenario, and listing extension condition and process steps. Finally, analysts extract complex steps as sub use case, and modify use case set.

Mark Collins-Cope presents a Requirments /Service /Interface approach [6]. This approch is mainly applied in business application systems, and it asks analysts to organize, construct and manage use cases from three levels (requirment, service and interface). Requirment use case describes business processes, not internal details of system. Service use case describes systemic basical functions. Interface use case describes interfaces between actors and related functions.

He Keqing etc. provide a role use case analysis method [7]. This method defines and analyzes use case from role's perspective, and captures system requirments by using role use case model. It also proposes the conbination of role use case and UML with the 3R (refinement, realization and reflection) mechanism and AOP.

Wei Dong etc. propose an actor-based use case analysis method [8].This method organizes and constructs use case from horizontal and vertical directions. First, analysts find actors by decomposing use case horizontally, and then write a series of actor-related interactive scenarios, and decompose use case vertically.

These methods build a bridge on the gap between analysis and design, and to some extent, improve use case analysis method. But they fail to solve some problems on granularity of use case.

3.3 PRINCIPLES AND STANDARDS

By current methods and experience, it is easy to acquire the following principles and standards.

(1)Take actor as the center. It asks analysts to capture requirments from actor's perspective, and constantly emphasize actor's goals. So selecting right actor is important. For different actors, it is better to analyze their related use cases separately.

(2)Gradually decompose, refine and revise use case. Analysts may first decompose them and get sketch, then refine them gradually, and finally check them by principles and standards and revise them.

(3)Use cases in the same abstract level, should maintain in high cohesion and low coupling condition. It is better to reduce relationships between use cases of different and same actors, and do not use relationships of use case until the use case model is relatively completed [9].

(4)In same requirment phase, granularity of use cases should be in same magnitude.

4. AN EFFECTIVE ANALYSIS METHOD

According to concepts, current methods and priciples, it proposes a feasible analysis use case method, and take a simple library system for example.

Firstly, analysts obtain business goals of system, clear who will be served, gain system scope, and find all actors according to users' features. Business goal of library system is managing books, and this business serves to readers and librarians. So system scope is library system, and Fig.1 gives the actors.

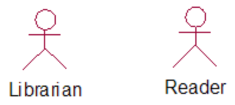

Fig.1 System actors

Secondly, analysts list all supportive actors and their duties, decompose their tasks by a series of interactions with the system, and then gain the use cases holding primary accuracy. The use case has no details, just expresses business requirments. It is appropriate for granularity of use case to describe a complete business process. Fig.2 gives the use cases about library system in this step.

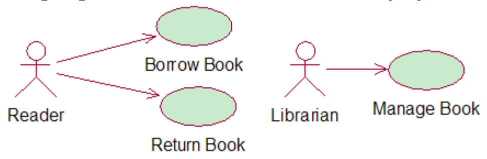

Fig.2 Use cases holding primary accuracy

Thirdly, analysts find main business use cases and analyze them by object-oriented method. Then analysts select several core work units, refine them gradually and attain a set of use cases having small granularity. It is appropriate for granularity of use case to describe a complete event flow, that is, one use case describe one step of a business. Refining process is as shown in Fig.3~Fig.5. By analyzing refined use cases, analysts draw scenarios by sequence diagram and activity diagram. By analyzing entity objects from systemic perspective, analysts get the relationships between analysis classes. Analysis class is the most important element for transform from business requirements to systen design, and it has three types, boundary class, control class and entity class. All operations are carried by boundary class, all operation messages are passed to and executed by control class from boundary class, and the datas are stored in entity class. Fig.6 presents a sample.

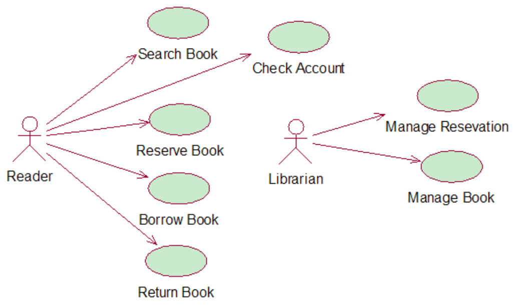

Fig.3 Firstly refine use cases

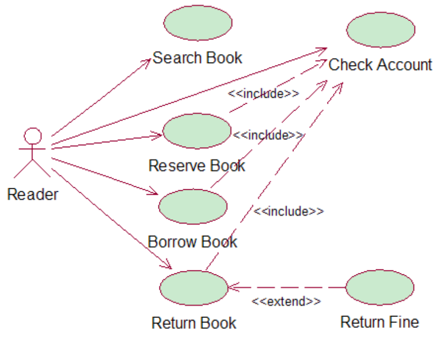

Fig.4 Further refine reader's use cases

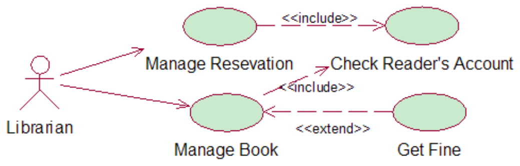

Fig.5 Further refine librarian's use cases

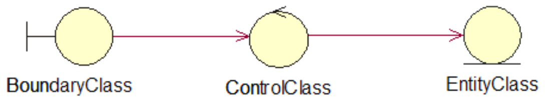

Fig.6 An analysis class sample

This step is a transitive process. It reflects original requirments upwards, and provides a high level abstraction downwards.

Finally, analysts decompose use cases from computer's perspective, which makes the transform from business requirments to computer program. Analysts extract available use cases, and bring them in the system by reflection, abstraction, combination, split and deduction method. Then analysts describe how computer deal with them and how operater operate them. It is appropriate for granularity of use case to describe a complete interaction between operator and computer. Fig.7 gives a use case digram of CheckAccount. In this step, boundary class, control class and entity class are separately transformed to operation view or system interface, computer program and database table, XML document, or other persistent class, as shown in Fig.8.

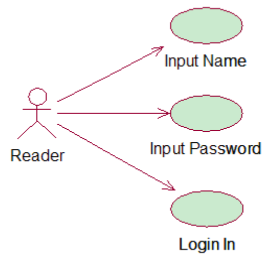

Fig.7 CheckAccount use case

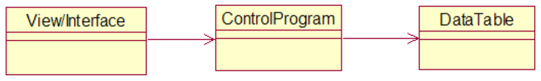

Fig.8 Describe use case from computer perspective

Though above steps, it is easy to draw particular use cases, and prepare for following design.

5. SUMMARIES

In requirment phase, analyzing use case is important. Inappropriate use case sets can bring a disaster for discerning systemic objects, get an irrational object construction, and finally affect the realization of system function and performance. According to commom mistakes and principles, it proposes an analysis use case method which is easy to understand and grasp. This method does not aim at special development scenarios, but commom system development. Future research will focus on practice of this method. By tracking practice process, collecting datas and measuring results, analysts can summarize pros and cons of the method.

REFERENCES

[1] Kurt Bittner; Ian Spence. Use case modeling[M]. Beijing: Tsinghua University Press, 2003, PP:04-05.
[2] Ivar Jacobson; Grady Booch; James Rumbaugh. Unified software development process[M], Beijing: Tsinghua University Press, 2005.
[3] Huang Yutian; Nie Liqin; Duan Fu etc. Application of use case requirment analysis technology, Journal of Taiyuan University of Technology[J], 2005.3, PP:224-227.
[4] Pan Jiayu. Does use case have granularity, Programer[J], 2008.3, PP:72-74.
[5] Alistair Cockburn. Writing effective use cases [M]. Beijing : China Machine Press, 2002.
[6] Mark Collins-Cope. The RSI Approach to Use Case Analysis, C+ + Reportor[J], 1999, PP:28-33.
[7] He Keqing; He Fei; Ying Shi. Role Use Case: A More Complete Analysis Method for UML, Journal of Computer Research and Development[J], 2001.9, PP:1105-1111.
[8] Wei Dong; Wen Dengmin; Wu Min. A Actor-Based Use Case Analysis Method, Computer Applications[J], 2004.6, PP:334-336.
[9] Yuan Tao; Kong Leilei. Summary of Researh and application of Use Case, Forest engineering[J], 2007.9, PP:85-88.