

Functional Specification Document Generation from Business Requirement

Patsuda Vittayanonaketavee, Prabhas Chongstitvatana
Chulalongkorn University
Faculty of Engineering
254, Phayathai Rd, Wang Mai, Pathum Wan, Bangkok 10330, Thailand
6070950321@student.chula.ac.th, prabhas.c@chula.ac.th

Abstract— This work focuses on automatic generation of Functional Specification Document from Business Requirement. In the requirement analysis process of software development, Business Analyst must gather needs from users, designs functions and writes many documents. To improve the consistency and reduce the discrepancy between requirement and the output specification documents, an automatic method is proposed. The proposed method generates Functional Specification in the form of Sequence Diagram. The system employs a template which contains Actor, Message, Object and Callback. A business analyst must fill in the additional information concerning subprocesses and states. The results from case studies show that the proposed method generates documents and sequence diagrams correctly.

Keywords— functional specification, business requirement, sequence diagrams

I. INTRODUCTION

In a software development process, there are many documents that are required before the system go live. Business Analyst (BA) must gather requirements from users, designs functions and makes sure all documents are finished on time. BA must handle many jobs, so there will be some misunderstanding about the requirements. One of the documents is Functional Specification (FS). It would be better if Functional Specification can be generated from Business Requirement (BR) before developing the system. This will make sure that they are consistent. This work proposes a set of rules and a working prototype that allows Functional Specification to be generated automatically from Business Requirement. It will help users to have an overview of the system requirements before starting to develop software.

II. BACKGROUND

A. Natural Language Processing (NLP)

NLP [1] is the processing of human language using computer logic to enable computers to understand human language according to the purpose of tasks or applications. The field has been developed since the late 1940s. Machine translation (MT) is the first computer program that has applications related to human language conversion. Liddy (1998) and Feldman (1999) suggest that to understand human language, it is necessary to raise the difference between seven levels that people use to extract meaning from text or spoken language [2] as follows.

- Phonetic or phonological level
- Morphological level
- Lexical level
- Syntactic level

- Semantic level
- Discourse level
- Pragmatic level

B. Unified Modeling Language (UML)

UML is a diagram used to represent programming concepts and shows details and describes the operation of the system. It is useful for visualizing the system before developing software. There are 13 types of diagrams that can be grouped [3] as follows.

- Structure diagrams are diagrams showing the structure, components and relationships of components in the system, including Class diagram Component diagram and others.
- Behaviour diagrams are behavioural diagrams of the system. It describes the functions of each component in the system, including use case diagram, activity diagram and state machine diagram.
- Interaction diagrams are part of the behavioural diagram but emphasizing the flow of information and communication sequences in the system, including sequence diagram and communication diagram.

C. Business Requirement Document (BR)

BR is a guideline for system design in the software development process. It is created by analysing the needs from the existing system and the system that users want to have in the future [4]. The document is prepared by the system analyst, users and stakeholders. The document usually consists of the following content.

- Scope and business details, including causes of changes
- Business people involved who have business needs
- Factors for the success of future goals
- Business and system limitations
- Map of the business process model of the current system and the system in the future
- Data model and data definition
- Definition of business terms
- Data flow diagrams

D. Functional Specification Document (FS)

FS is the document that describes the components and functions in the system [5]. It is used to help system

developers to understand the requirements and to develop the system that meet the needs of users.

III. LITERATURE REVIEW

There are numerous work to generate UML from text. For example, [6] generates use case and class diagram. They extract key features from text and use machine learning for classification. The work in [7] generates UML class from software requirement. A vocabulary repository is used to organise classes and relationships. Natural language processing approach has been used to resolve uncertainty in requirement [8]. It detects and corrects ambiguity using WordNet. UML components is identified and the diagram is generated. Use cases can be generated from user-stories using NLP [9]. NLP is used to separate use-stories into words. Actor is extracted from noun. A verb is identified and is used in use cases. NLP is used to analyse and convert requirement to UML diagram [10]. OpenNLP parser is used to identify the type of words. The relationships are extracted and then used to generate UML diagram.

IV. PROPOSED APPROACH

A proposed document generator aims to generate Functional Specification Document from Business Requirement. The program is developed by using Java language and using API¹ to generate sequence diagrams. The generator has the following steps. First, extract words from the requirement². Noun is identify as Actor. Verb is identify as Message. The template is designed to specify what is Actor, Message, Object and Callback. Those are the structure of sequence diagram.

Rules to extract words from requirement

- 1) There is Mylist to identify verbs berfore Actor.
- 2) Loop to separate words and sentences. Then, collecting sentences in Finalist
- 3) Loop to find Actors from Finalist.
 - 3.1. Words are cut by dictionary
 - 3.2. Loop, take the words, check with Mylist.
 - 3.3. If the word is not in mylist, continue with the same word.
 - 3.4. If present, hold the previous word as Actor and put it in actorlist
 - 3.5. break and make the next sentence

After inserting Actors to the template, the entire information of the template is fulfilled by BA. Then, the generator converts the template in to script by following rules.

Rules to convert template to script

- 4) Loop to add List of Actor, Message, Object and Callback by column index
 - 4.1. If column index = 0 then add Actor
 - 4.2. If column index = 1 then add Message
 - 4.3. If column index = 2 then add Object
 - 4.4. If column index = 3 then add Callback
- 5) Loop to write script
 - 5.1. If Actor column is not space and Message column is not space and Object column is not space then

Actor+ '->' + Object+ ':' + 'Message' + newline;

5.2. If Callback column is not space then

Object+ '->' + Object+ ':' + 'Message' + newline;

6) Generate output from the script by calling getSequenceDiagram() and identify output path

Finally, Function Name is used to map contents to parse sequence diagram in each function and parse the details in Functional Specification from Business Requirement.

To illustrate the working of the generator steps, we will use one example. Here is the Business Requirement of the Monitor Function.

“Monitor Function is used to present logs in page and authorize user who can access to logs. User will be validated with BankList and ActionList. Then, present the validation result back to Monitor Page. User can search for logs that are required to be monitored on MonitorPage. User can search and get only their own transaction logs. Only Administration can get all logs. Rights will be validated by ADConnector.”

Table I shows the template that is filled in by the rules 1-4 from the Business Requirement.

TABLE I. MONITOR FUNCTION

Actor	Message	Object	Callback
User	Call	Monitor Page	-
Monitor Page	Validate BankList	BankList	-
BankList	Send Items	Monitor Page	-
Monitor Page	Validate Action	ActionList	-
ActionList	Send Items	Monitor Page	-
Monitor Page	Present	User	-
User	Call Logs	Monitor Page	-
Monitor Page	GetLogs	Collector	-
Collector	GetUsers	ADConnector	-
ADConnector	ListUser	Collector	GetFileName ByDate
		Collector	GetFileLogs
Collector	LogView	Monitor Page	-
Monitor Page	Present	User	-

Script form

Once the template is completed, a script is generated as shown by rules 5 and the sequence diagram is generated from this script (rule 6). The final sequence diagram is generated from this script.

User->Monitor Page: Call
 Monitor Page->BankList: Validate BankList
 BankList->Monitor Page: Send Items

¹ API from www.websequencediagrams.com

² Using pythainlp

Monitor Page->ActionList: Validate ActionList
 ActionList->Monitor Page: Send Items
 Monitor Page->User: Present
 User->Monitor Page: Call Logs
 Monitor Page->Collector: GetLogs
 Collector->ADConnector: GetUsers
 ADConnector-> Collector: ListUser
 Collector->Collector: GetFileNameByDate
 Collector->Collector: GetFileLogs
 Collector-> Monitor Page: LogView
 Monitor Page->User: Present

Output: Sequence diagram of Monitor Function

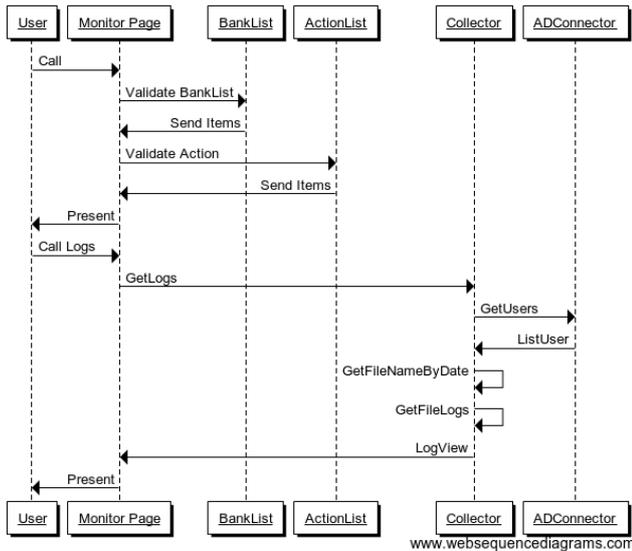


Fig. 1. Sequence diagram of Monitor Function

V. CASE STUDIES

Two cases are presented to demonstrate the proposed method. The first case is Password Management. The second case is Password Setting. The Business Requirement for both cases are straightforward.

Case 1: Password Management

“Password Management is a page that is used to set permissions to users to access Password Setting Page and authorized user who can access the page.”

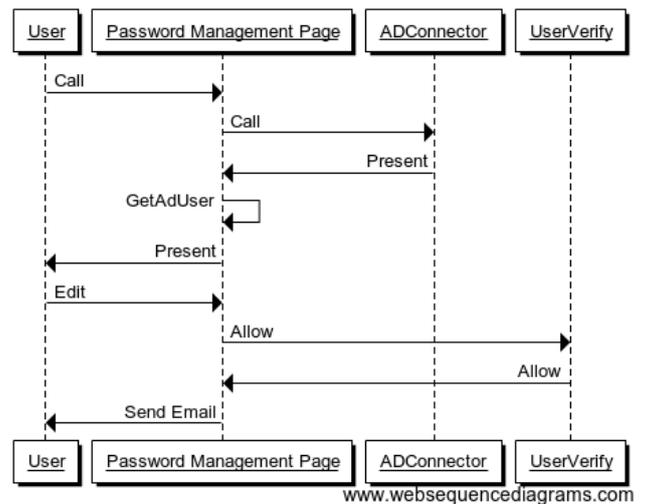
TABLE II. PASSWORD MANAGEMENT FUNCTION

Actor	Message	Object	Callback
User	Call	Password Management Page	-
Password Management Page	Call	ADConnector	-
ADConnector	Present	Password Management Page	GetAdUser
Password Management Page	Present	User	-
User	Edit	Password Management Page	-
Password Management Page	Allow	UserVerify	-

Actor	Message	Object	Callback
UserVerify	Allow	Password Management Page	-
Password Management Page	Send Email	User	-

To fill the template, the analyst must know the states of actions to accomplish the task. For example, the password page is connected to ADConnector. The details such as to let user edit password and verify it, are a kind of pattern for this specific function. Some extra function such as “email user” is added on by the analyst. These actions are reflected in the template. The output sequence diagram is generated from the template as shown below.

Output: Sequence diagram of Password Management Function



Case 2: Password Setting

“Password Setting is a page that used to set password before access to the system.”

In this case, there are many steps that a user must verify the password. First step, the request to change password may be denied or fail because of over time limit. The submitted password may be incorrect or fail because of over time limit. Second step, once the input password is accepted. It is sent to be validated. If it is the resetting password PrincipleManager will be invoked. These details are filled in by the analyst.

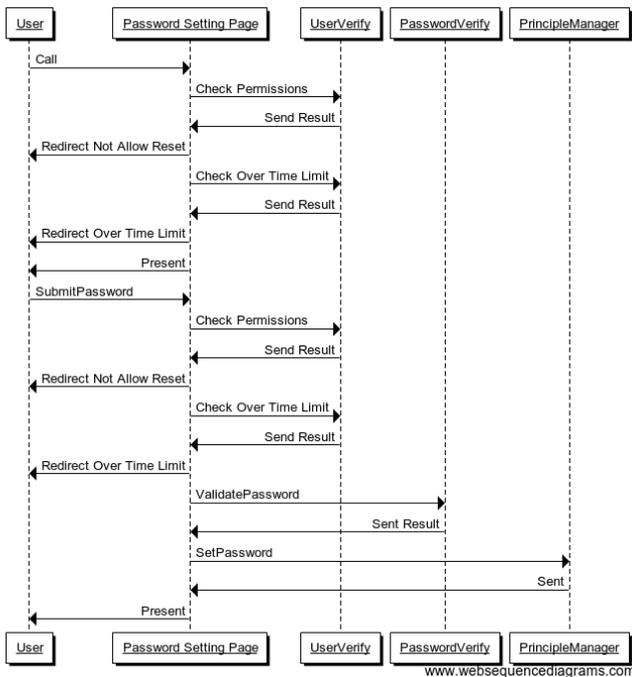
TABLE III. PASSWORD SETTING FUNCTION

Actor	Message	Object	Callback
User	Call	Password Setting Page	-
Password Setting Page	Check Permissions	UserVerify	-
UserVerify	Send Result	Password Setting Page	-
Password Setting Page	Redirect Not Allow Reset	User	-
Password Setting Page	Check Over Time Limit	UserVerify	-
UserVerify	Send Result	Password Setting Page	-
Password Setting Page	Redirect Over Time Limit	User	-
Password Setting Page	Present	User	-

Actor	Message	Object	Callback
User	SubmitPassword	Password Setting Page	-
Password Setting Page	Check Permissions	UserVerify	-
UserVerify	Send Result	Password Setting Page	-
Password Setting Page	Redirect Not Allow Reset	User	-
Password Setting Page	Check Over Time Limit	UserVerify	-
UserVerify	Send Result	Password Setting Page	-
Password Setting Page	Redirect Over Time Limit	User	-
Password Setting Page	ValidatePassword	PasswordVerify	-
PasswordVerify	Sent Result	Password Setting Page	-
Password Setting Page	SetPassword	PrincipleManager	-
PrincipleManager	Sent	Password Setting Page	-
Password Setting Page	Present	User	-

The output sequence diagram is generated from the template as shown.

Output: Sequence diagram of Password Setting Function



ACKNOWLEDGMENT

The author is thankful to people of IT department those provided helpful information and data.

REFERENCES

- [1] Elizabeth D. Liddy, Natural Language Processing, Syracuse University, 2001.
- [2] Gobinda G. Chowdhury, "Natural Language Processing," Annual Review of Information Science and Technology 37(1):51-89, 2005.
- [3] Object Management Group., "Unified Modeling Language (UML)", 2014.
- [4] Goldsmith, Robin F., Discovering Real Business Requirements for Software Project Success, Artech House, 2004.
- [5] ISO/IEC/IEEE 24765-2010, 2010
- [6] Chamitha Narawita, Kaneeka Vidanage, "UML Generator – Use Case and Class Diagram Generation from Text Requirements", Int. J. on Advances in ICT for Emerging Regions, 2018.
- [7] Hina Afreen, Imran Sarwar Bajwa, "Generating UML Class Models from SBVR Software Requirements Specifications", Benelux Conference on Artificial Intelligence, 2011.
- [8] Gaurav A. Patel, A Swathy Priya, "Resolve the Uncertainty in requirement Specification to generate the UML Diagram", Int. Conf. on Advances in Engineering and Technology, 2014.
- [9] Ahmad Azzazi, "A Framework using NLP to automatically convert User-Stories into Use Cases in Software Projects," Int. J. of Computer Science and Network Security, Vol.17, No.5, 2017.
- [10] S.D.Joshi, Dhanraj Deshpande, "Textual Requirement Analysis for UML Diagram Extraction by using NLP," Int. J. of Computer Applications, Vol. 50, No.8, 2012.

VI. CONCLUSION

This work proposes a method to generate Functional Specification from Business Requirement. From case studies presented, they were found that the production of document can be used in practice. The generator helps both users and developers to have mutual understanding of the target system. The generator also help the analyst to formally define components and objects so that all the hidden states can be presented explicitly.