

Functional Requirements Identification Using Item-to-Item Collaborative Filtering

Reynald Jay F. Hidalgo and Proceso L. Fernandez

Abstract—One of the most difficult tasks in the development of software is the identification of the functional requirements. A well-defined functional requirement will eventually map the success of a software project. A support tool that can recommend candidate functional requirements for a software project being developed will help software engineers to deliver the right software to the clients.

The purpose of this study is to determine whether a collection of previously developed software applications can serve as basis for the development of a model to identify functional requirements of a project to be developed. Completed software project documentations of Master in Information Technology graduates from Higher Education Institutions in northern Luzon were collected and categorized, and projects on Faculty Evaluation Systems were analyzed in order to extract the vectors of functional requirements for the target software domain. Item-to-item collaborative filtering is then applied to these vectors to form the recommender model. Finally, the model was validated using the leave-one-out cross-validation technique. The resulting high accuracy rate indicates that the model built using the item-to-item collaborative filter may be used to build a tool that can automatically recommend important related functional requirements given one or more requirements identified by any of the project stakeholders.

Index Terms—Collaborative filtering, functional requirements, item-to-item collaborative filtering, requirements engineering.

I. INTRODUCTION

The software industry constantly increases the demand for functionality. Functional requirements sketch out the functionality the system has to perform [1]. They define what the system will do specifically in the context of an external interaction either with a user or with another system. The importance of requirements in general and the underlying problems that goes with it had been the subject of several surveys since the early 90's. As systems becomes more multifaceted, software developers are given very little assistance on how to overcome the difficulties that naturally come with such complex software projects. Several of the surveys have highlighted principal flaws in the requirement process that can be linked to accidental difficulties such as tool integration [2]. In a survey on market-driven requirements engineering conducted by Karlsson *et al.* (2002), one of the salient challenges found is the absence of

simple tools for basic needs. [3] The respondents of the said survey requested simple and easy to use techniques for basic activities in requirements engineering. Cheng and Atlee (2007) identified that much effort has been devoted to leverage tools and technologies enhancing requirements elicitation but there still exist a vast potential for improvement [4].

The most obvious documentation tool used is a word processing packages which is labelled as a poor documentation tool [5]. The same issue was raised in 2000 by Nikula *et al.* who found out that no software organization used requirements management tools. Only templates, checklists, and metrics were used by a handful of companies to document their requirements while some companies had tools that aid them for the other software phases that includes configuration management and testing [6]. The presence of tools to help developers for organizing and documenting various parts of the software development process is critical to the success of a project.

The most important element in the identification of the functional requirements is the stakeholders. This research attempts to answer the question, "Can collaboration among multiple data sources (of previously created software projects) identify functional requirements of stakeholders in new software projects to be developed?"

II. CONCEPTUAL FRAMEWORK

We hypothesize that using item-to-item collaborative filtering applied to completed software engineering projects can model a requirement elicitation tool that can recommend functional requirements to projects to be developed. Fig. 1 shows the conceptual framework of this research.

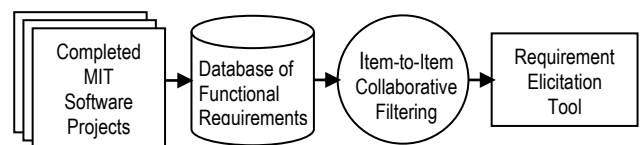


Fig. 1. Conceptual framework.

Previously developed systems can be a good source of functional requirements for projects currently being developed. McMEn-amin and Palmer [7] argued that new software projects have to meet a large fraction of the functionality of the old system; and that errors can be avoided based from the lessons learned from the experiences of completed projects. Furthermore, functional requirements which are not properly identified at the onset of the project development may sacrifice cost and high quality of requirement. Reviewing the past developed projects can

Manuscript received May 15, 2014; revised July 23, 2014.

Reynald Jay F. Hidalgo is with School of Information Technology Education of Colegio De Dagupan, Dagupan City, Philippines (e-mail: reynaldjayhidalgo@yahoo.com).

Proceso L. Fernandez is with Ateneo de Manila University, Quezon City, Philippines.

contribute to cost efficiency and early removal of defects [8].

III. RELATED WORK

A. Requirements Elicitation

Requirements identification is one of the tasks in the requirements engineering phase in the software development cycle. The lack of tools that will aid software engineers properly enumerate the right requirements had been the subject of recent studies, mostly utilizing concepts in artificial intelligence to elicit the right requirements.

Asadi *et al.* [9] developed a feature model configuration approach that takes into account qualitative and quantitative non-functional properties as well as functional requirements. At the same time, capturing the preferences of stakeholders in terms of relative importance between non-functional properties is also included. This can make it easier for stakeholders to identify these requirements. A combination of Analytical Hierarchy Process (AHP) and Fuzzy Cognitive Maps (FCM) are used to compute the non-functional properties weights based on stakeholders' preferences and interdependencies between non-functional properties. Furthermore, Hierarchical Task Network (HTN) planning is applied to find the optimal feature model configuration that will eventually used to automatically select proper features based on stakeholders' requirements.

The importance of identifying the right requirements is cascaded even to very specific software domains as this eventually contribute to the development of software that is accepted by end-users.

This had been the subject of the study of Hussin and Johare (2014) [10] who investigated the necessity of identifying the functionality of a record-keeping system, the anticipated user characteristics, and how functionality can be kept abreast of ever-changing technology platforms. They applied this concept in the development of an electronic court records management system and had proven that functional requirements are essential in the completion of the said system.

Casamayor, Godoy and Campo (2009) [11] developed a semi-supervised text categorization approach for the automatic identification and classification of non-functional requirements. The goal of the approach is the integration into a recommender system that will assist software engineers in the development of the system. This approach uses a few requirements identified by software engineer during the requirements elicitation phase, and allow learning an initial classifier for the non-functional requirements that could successively identify the type of requirement in an iterative process.

B. Item-to-Item Collaborative Filtering Algorithm

Collaborative filtering had been proven useful in the development of recommender systems. Recommendation algorithms are best known for their use on e-Commerce Web sites, where they use input about the customer's interest to generate a list of recommended items. Perhaps the most famous retailer who applied this concept is Amazon.com where upon the purchase of item, provides recommendations

to possible items that a customer might want to buy [12].

In the field of education, this technique was used to develop a web-based intellectual e-notebook system [13]. This system can recommend the useful notes automatically based on the item-to-item collaborative filtering applied to the learners' reading histories and contents of notes, which improves the efficiency of group learning.

Other applications of this algorithm include a distributed semantic p2p overlay which can provide music search and recommendation services by considering both of user preference and diversity of interest [14]. The study combined item-to-item relationship to construct a history based cooperative filter.

In recent years where there has been an increase in online video content, this has given rise to the need of developing effective recommendation system which can help users discover meaningful and interesting videos. Jin *et al.* (2010) used item-to-item collaborative filtering using video views data, which involves content analysis of video metadata to extract feature representation for identifying similar videos for recommendation. Recommendation results were further filtered through a refinement stage using semantic similarity [15].

Recommender systems are also being used in web services to identify non-functional requirements. A graphical user interface was even included to facilitate ease of use of the tool by the different users of the system and the user can specify their requirements for desired web services [12].

IV. METHODOLOGY

A. Data Collection

Software project documentations are used as basis for building the model that will identify possible functional requirements of a software project currently being developed. These project documentations, alongside the software, are written by students as a requirement for graduation to the Master in Information Technology (MIT) program. Capstone projects serve to provide the MIT students the opportunity to apply the concepts they have learned from the academic courses they finished. This is offered during the last semesters that the student is enrolled prior to graduation and capstone projects are normally developed within a span of one year.

Every MIT student is required to submit a project for the completion and awarding of the MIT degree. These projects are defended by the student to a panel of experts. The members of the panel are selected from a pool of experts based from the relevance of the project being presented to the expert's specialization. The major responsibility of the panel is to ensure that the projects had met the requirements of the target beneficiaries

All projects and their corresponding documentations that were successfully defended are placed in the libraries of the institutions that offer the program. These project documentations are strategically placed in the libraries to serve as basis and reference materials for MIT students who are currently developing their software and consequently writing their software project documentation. These

documents more or less describe the system developed, the functionalities that are included and the processes that were executed by the MIT student. Essential elements such as the domain, the stakeholders, and the functional and non-functional requirements are also included in the software project documentation.

In this study, purposive sampling technique was used to select the documents to be collected and evaluated. MIT projects from graduate schools offering this program in Northern Luzon, Philippines are used in the study. These documents include completed software project documentations from Colegio De Dagupan, Don Mariano Marcos Memorial State University and the University of the Cordilleras. Permission to access the said documents was secured from the pertinent officials of the concerned Higher Education Institutions prior to gathering them.

B. Data Analysis

A review of completed projects through document analysis is used to extract the profile along software domain, stakeholders, and functional requirements. Document analysis describes the act of reviewing the existing documentation of comparable business processes or systems in order to extract pieces of information that are relevant to the current project. Existing documentations is reviewed for understanding key functional requirements, the domain of the project and the classification of stakeholders.

Specifically, the parts of the documents that are analysed included the Discussion of Findings, Appendices that includes Use-Case diagrams, and transcripts of the interview among others. This serves as the basis in the identification of functional requirements, stakeholders, and the domain. The data gathered from these projects are used further to develop a tool to help software engineers to identify functional requirements.

C. Selection of Target Software Domain

All project documentations developed at the onset of offering the program of the identified HEIs are included in the study. The projects are grouped on the basis of the domain of the software, and a specific domain is selected for the development and validation of the recommender model.

D. Model Building

The item-based collaborative filtering applied in purchase statistics shall be used in this study. In this method, the users are only provided with binary data. One (1) will represent an item that was purchased while zero (0) indicates that the item that was not purchased. This is very similar to the binary item-based collaborative filtering patented algorithm by Amazon.com [10]. Table I shows a sample purchase statistics.

TABLE I: SAMPLE PURCHASE STATISTICS FOR ITEM-TO-ITEM COLLABORATIVE FILTERING

Customer	Item 1	Item 2	Item 3
1	Bought it	Didn't buy it	Bought it
2	Didn't buy it	Bought it	Bought it
3	Didn't buy it	Bought it	Didn't buy it

In this case, the cosine between items 1 and 2 is:

$$(1, 0, 0) \cdot (0, 1, 1) = 0$$

$$\|(1, 0, 0)\| \|(0, 1, 1)\|$$

The cosine between items 1 and 3 is:

$$(1, 0, 0) \cdot (1, 1, 0) \approx 0.71$$

$$\|(1, 0, 0)\| \|(1, 1, 0)\|$$

Whereas the cosine between items 2 and 3 is

$$(0, 1, 1) \cdot (1, 1, 0) \approx 0.5$$

$$\|(0, 1, 1)\| \|(1, 1, 0)\|$$

E. Model Validation

In this paper, leave-one-out cross-validation (LOOCV) is used to estimate the expected accuracy of the model. Though statistical accuracy metrics evaluate the accuracy of an identifier by comparing predicted values with user provided values, the existing data instead was used for this purpose. One project will serve as the test data while all the rest of the existing projects serves as the basis of the model. All the projects included will eventually serve as test data at one time and as part of the model for n-1 time where n is the total number of projects under the identified domain. As an example, Project 1 will serve as test data while Project 2 to Project n will be used as the basis of the model. The requirements will then be subjected to the model and each time a functional requirement is present on Project A, all functional requirements that will be identified by the model will be recorded. The recorded values will then be sorted from highest to lowest. The top n-1 shall be extracted from the list. These functional requirements shall then be compared with the actual functional requirements in the project being tested. If the recommended functional requirement is present from the actual functional requirement, then this will be a hit. All hits shall then be added. The total hits of all the actual functional requirements shall be added divided by the total number of functional requirements. The result will then be the average hits of all the requirements out of all the n functional requirements of the project being tested.

V. RESULTS

A. Profile of Completed Projects

The three HEI's offering MIT in Northern Philippines had yielded a total of 316 projects. The most number of projects developed is from the University of the Cordilleras which totalled to 199, followed by Colegio De Dagupan which totalled 59 and DMMMSU which totalled to 58. It is expected that the University of the Cordilleras had produced the largest number since they had been operating the MIT program since 2006 while the two other institutions started operation only last 2008. The University of the Cordilleras had also been identified by CHED as a delivering institution in government funded scholarship grants for college faculty. Although Colegio De Dagupan was also identified as a delivering institution the moment they started operation. DMMMSU on the other hand is state-run university offering lower fees as compared to the two other private institutions. Table II shows the top 15 projects that are produced by the

three HEIs.

TABLE II: TOP 15 PROJECTS BASED FROM SOFTWARE DOMAIN

RANK	DOMAIN	UC	CdD	DMNMSU	TOTAL
1	eLearning Systems	28	5	4	37
2	Human Resource Information System	16	5	3	24
3	Portal	6	1	11	18
4	Student Information System	10	4	3	17
5	Inventory System	6	7	2	15
6	Faculty Evaluation System	10	2	1	13
7	Library Management System	4	6	2	12
7	Alumni Tracer System	6	2	4	12
9	Student Grading System	6	2	2	10
10	Document Management System	2	3	2	7
10	Enrolment System	3	2	2	7
12	Academic Advising System	3	2	1	6
13	Patient Records Management System	5	0	0	5
14	Examinations System	3	1	0	4
15	Attendance Monitoring	2	0	1	3

Faculty Evaluation Systems, ranked six on the top 15 list is used as the subject of the model. The number of completed project is just right for the development of the model. There are 13 projects under this software domain.

Table III shows the profile of the chosen domain and the stakeholders who are the primary source of the functional requirements.

TABLE III: STAKEHOLDERS UNDER THE DOMAIN OF FACULTY

Domain	Stakeholder/s
Faculty Evaluation System	Students Dean Peers Guidance Counsellors System Administrators

TABLE IV: FUNCTIONAL REQUIREMENTS BASED FROM THE COMPLETED SOFTWARE PROJECTS

No.	Functional Requirements
1	Answer evaluation
2	View evaluation criteria
3	Add comments
4	Save answers
5	Check completeness of response
6	Manage Log In/Out
7	Check evaluation schedule
8	Allow on-line help
9	Search faculty to be evaluated
10	Add evaluation criteria
11	Update evaluation criteria
12	Edit evaluation criteria
13	Delete evaluation criteria
14	Compute evaluation response
15	Generate evaluation result
16	Generate descriptive report
17	View results
18	Print results
19	Create user account
20	Perform system audit
21	Create schedule
22	Check list of evaluators
23	Create backup
24	Archive database content
25	Restore database content
26	Connect to Existing Database

These stakeholders had either answered a survey questionnaire floated by the project study developers and/or had scheduled for a structured interview. In Faculty Evaluation Systems, the Students, the Dean and Peers evaluate a faculty member. On the other hand, all of the institutions that were selected as the object of study of the project developers had in-house system administrators that were also interviewed.

B. Analysis of the Profile of the Projects Using the Item-to-Item Collaborative Filtering

Table IV shows the functional requirements identified by the stakeholders of Faculty Evaluation Systems developed by MIT students.

The numbers on the latter table will be used to reference the functional requirements on the next table. Table V shows the presence of all the combined functional requirements from all the documents that were analyzed.

TABLE V: PRESENCE OF THE IDENTIFIED FUNCTIONAL REQUIREMENTS ON EACH OF THE DEVELOPED FACULTY EVALUATION SYSTEMS (RATINGS TABLE)

PROJECT	FUNCTIONAL REQUIREMENTS																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
A	1	1	1	1	0	1	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0
B	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
C	1	1	1	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	0	1	0	1	0	1	1
D	1	1	1	1	1	0	0	1	0	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0
E	1	1	0	1	0	1	1	0	0	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1
F	1	1	1	1	1	1	0	0	1	0	0	1	1	0	1	1	1	1	1	0	1	1	0	0	0	0
G	1	1	1	0	1	1	0	0	1	1	1	0	1	1	0	1	1	1	1	0	1	0	0	1	1	0
H	1	1	1	1	1	0	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0
I	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1
J	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
K	1	1	1	0	1	0	1	0	1	1	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0
L	1	0	1	1	0	1	0	0	1	0	0	0	1	1	0	1	1	1	1	1	0	1	1	1	1	0
M	1	1	1	1	1	0	1	0	1	0	0	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0

The table shows the presence of an identified functional requirement. One (1) denotes presence of the requirement while zero denotes its absence. Item-to-Item filtering was then applied to predict possible requirements that one software clients might seek to integrate in the software he wants. The item-to-item collaborative filtering algorithm is applied to the matrix to get the cosine values of each of the pairs. Table VI shows the functional requirements with the highest cosine values.

TABLE VI: FUNCTIONAL REQUIREMENTS WITH HIGHEST COSINE VALUES

Functional Requirement	Cosine Values																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26				
1		0.96	0.96	1.00	0.73	1.00	0.73	0.55	0.88	0.88	0.68	0.55	1.00	1.00	0.62	1.00	1.00	1.00	0.62	0.92	0.62	0.55	0.55	0.55	0.55	0.55				
2			0.92	0.96	0.76	0.96	0.65	0.58	0.82	0.91	0.71	0.58	0.96	0.96	0.65	0.96	0.96	0.96	0.65	0.87										
3				0.96	0.76	0.96	0.65	0.58	0.82	0.82	0.71	0.58	0.96	0.96	0.65	0.96	0.96	0.96	0.65	0.87										
4					0.73	1.00	0.73	0.55	0.88	0.88	0.68	0.55	1.00	1.00	0.62	1.00	1.00	1.00	0.62	0.92	0.62	0.55	0.55	0.55	0.55	0.55				
5						0.73	0.57	0.76	0.60	0.60	0.57	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.68	0.57										
6							0.73	0.55	0.88	0.88	0.68	0.55	1.00	1.00	0.62	1.00	1.00	1.00	0.62	0.92	0.62	0.55	0.55	0.55	0.55	0.55				
7								0.72	0.72												0.68	0.68	0.57	0.57	0.57	0.57				
8																														
9														0.55	0.55		0.55	0.55	0.55	0.89										
10										0.90	0.77	0.63	0.88	0.88	0.71	0.88	0.88	0.88	0.88	0.95	0.57	0.63	0.63	0.63	0.63	0.63	0.63			
11											0.77	0.63	0.88	0.88	0.71	0.88	0.88	0.88	0.88	0.57	0.86									
12												0.82	0.68	0.68	0.73	0.68	0.68	0.68	0.68	0.74										
13													0.55	0.55	0.67	0.55	0.55	0.55	0.67	0.60										
14														1.00	0.62	1.00	1.00	1.00	0.62	0.92	0.62	0.55	0.55	0.55	0.55	0.55	0.55			
15															0.62	1.00	1.00	1.00	0.62	0.92	0.62	0.55	0.55	0.55	0.55	0.55	0.55			
16																0.62	0.62	0.62	0.62	0.67	0.67						0.63			
17																	1.00	1.00	0.62	0.92	0.62	0.55	0.55	0.55	0.55	0.55	0.55			
18																		1.00	0.62	0.92	0.62	0.55	0.55	0.55	0.55	0.55	0.55			
19																			0.62	0.92	0.62									
20																														
21																						0.67	0.60	0.60	0.60	0.60	0.60			
22																							0.67							
23																								0.75	0.75	0.71	0.71	0.71		
24																										1.00	1.00	1.00		
25																														
26																														

A stakeholder who identified functional requirement number 7 has a high tendency that he may also want to include functional requirements 10, 11, 14, 15, 17, 18, 19, 21, 22, 23, 24 and 25. A stakeholder who identified functional requirement number 21 may not want functional requirements' number 22, 23, 24 and 25 as part of the system he wants developed.

C. Validation of the Model

Each of the projects will be used at one time as test data while all the rest will be used to design the model. The functional requirement present on the test project is used to

get the recommendation of the model. From this list only $n-1$ shall be considered and compared to the actual functional requirements. Table VII shows the results of the validation of the model.

TABLE VII: NUMBER OF HITS OF THE FUNCTIONAL REQUIREMENTS PER PROJECT

PROJECT	FUNCTIONAL REQUIREMENTS																										AVERAGE HITS	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26		
A	13	13	14	13	13					13	13	14	15	13	13	13	13	14	13									13.3125
B	16	16	16	16	16	16	14	18		15	17	15	16	16	16	14	16	16	16	15								15.7895
C	14	14	13	14	14					17	14	14	14	14	14	14	14	14	14	14	14	16	15	15				14.3333
D	11	11	11	11	12	11		12						11	11	11	11	11	12	10	9							11
E	14	13	14	14	16					16	13			14	14	14	14	14	14	14	15	17	17	17	16			14.7368
F	14	13	14	14	12	14	14			13	13			14	14	14	14	14	14	14	11							13.5
G	14	14	14	14	14	13				14	14	13		14	14	14	14	14	14	14			13	13				13.7647
H	9	9	9	9	10	9								9	9	9	9	9	9									9.09091
I	21	20	20	21	22	21	21		22	21	20	20	22	21	21	20	21	21	21	21	22	20	22	22		22	21.0435	
J	16	18	17	16	17	16			16	14	17	16	17	16	16	14	16	16	16	16	16	14					16	
K	12	12	12	12	12	12	13	12	12					12	12	12	12	12	12	12	12						12.0714	
L	11	11	11	11	11	12			11					11	11	11	11	11	11	11	12	12	13	13	13		11.5625	
M	12	11	12	12	12	11	12		11					12	12	12	12	12	12	12							11.8	

Out of 16 functional requirements for project A, the model was able to correctly identify 13 given the functional requirement 1 while given functional requirement 13, the model was able to correctly identify 15 out of all the 16 requirements. The blanks on the table denote the absence of the requirement on that particular project. The average hits for project A is 13.3125. Furthermore, the model is able to recommend correctly 13.3125 out of the 16 actual functional requirements in Project A.

VI. CONCLUSIONS

Tools that will aid in requirements engineering process plays a vital role in the success of the software project. Item-to-item collaborative filtering had been proven to be one of the most successful techniques in the development of a recommender system. In this paper, a recommender model which uses item-to-item collaborative filtering to recommend functional requirements was successfully developed.

Results show that the functional requirements identified by the model are very close to that of the ones identified by the stakeholders of the projects. Keeping track of the functional requirements which were not identified by the stakeholder but is identified by the model are functional requirements noteworthy of inclusion in the project being developed.

REFERENCES

[1] S. Ullah, M. Iqbal, and A. M. Khan, "A survey on issues in non-functional requirements elicitation," in *Proc. 2011 International Conference on Computer Networks and Information Technology (ICCNIT)*, 2011, pp. 333-340.
 [2] E. Kamsties, K. Hornmann, and M. Schlich, "Requirements engineering in small and medium enterprises: state-of-the-practice, problems, solutions and technology transfer," in *Proc. Conf. European Industrial Requirements Engineering*, 1998, pp. 40-50.
 [3] L. Karlsson, B. Regnell, J. Karlsson, and S. Olsson, "Post release analysis of requirements selection quality- an industrial case study," in

Proc. 9th International Workshop on requirements Engineering: Foundation for software Quality, 2003, pp. 47-56.
 [4] B. H. C. Cheng and J. M. Atlee, "Research directions in requirements engineering," in *Proc. 2007 Future of Software Engineering (FOSE '07)*, 2007, pp. 285-303.
 [5] K. El Emam and N. H. Madhavji, "A field study of requirements engineering practices in information systems development," presented at IEEE 2nd Int'l Symp, 1995, Requirements Eng., pp. 68-80.
 [6] U. Nikula, J. Sajaniemi, and H. Kalviainen, *A State-of-the-Practice Survey on Requirements Engineering in Small Medium-Sized Enterprises*. 2000
 [7] S. M. McMenamin and J. F. Palmer, *Essential System Analysis*, Prentice Hall, 1984.
 [8] B. Xu, "Cost efficient software review in an e-business software development project," in *Proc. 2010 International Conference on E-Business and E-Government*, 2010, pp. 2680-2683.
 [9] M. Asadi, S. Soltani, D. Gasevic, M. Hatala, and E. Bagheri, "Toward automated feature model configuration with optimizing non-functional requirements," 2014.
 [10] N. Hussin and J. Rusnah, *Concepts and Advances in Information Knowledge Management Studies from Developing and Emerging Economies*, pp. 73-90, 2014.
 [11] A. Casamayor, D. Godoy, and M. Campo, *Identification of Non-Functional Requirements in Textual Specification: A Semi-Supervised Learning Approach*, 2009.
 [12] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *Internet Computing*, vol. P7, no. 1, pp. 76-80, 2003.
 [13] T. Ninomiya and T. Okamoto, "Development of an intellectual e-Notebook syste for group learning support," in *Proc. 7th International conference on Advanced Learning Technologies*, 2007.
 [14] M. Chen, K. C. Lin, C. Kung, and C. Tu, "On the design of the semantic P2P system for music recommendation," in *Proc. International Symposium on Parallel Processing with Applications (ISPA)*, 10.1109/ISPA, 2010, pp 442-448, 2010.
 [15] Y. Jin, M. Hu, H. Singh, and Z. Xie, "MySpace video recommendation with map-reduce on Qizmt," presented at 2010 IEEE Fourth International Conference on Semantic Computing 2010 IEEE International Conference on Semantic Computing, 2010.



Reynald Jay F. Hidalgo was born in Dagupan City, Philippines on May 1, 1976. He is a holder of BS computer science degree from Saint Louis University, Baguio City, Philippines in 1999, MS computer science degree at AMA University, Quezon City, Philippines in 2003 and doctorate degree in philosophy major in technology education management at DMMMSU, City of San Fernando, La Union, Philippines in 2011.

He is currently the director of Academic Affairs and concurrently the dean of the School of Information Technology Education of Colegio De Dagupan, Dagupan City, Philippines.

Dr. Hidalgo is a former vice president for Luzon of the Philippine Society of Information Technology Educators (PSITE) and former President of the Council of IT Deans and Heads (CITEDH) of Region 1.



Proceso L. Fernandez was born in Quezon City, Philippines, on June 19, 1972. He received the B.S., M.S. and Ph.D. degrees in computer science from the Ateneo de Manila University, Quezon City, Philippines, in 1994, 2001 and 2009, respectively.

He is currently an associate professor at the same university. He has been a visiting researcher at the Nara Institute of Science and Technology, Nara, Japan, since 2011. He has also been a visiting professor at the Ateneo de Davao University, Davao, Philippines and at the University of the Cordilleras, Baguio, Philippines.

Dr. Fernandez is currently a member of the Computing Society of the Philippines.