2010 Asia Pacific Software Engineering Conference

A Model and Architecture of REBOK(Requirements Engineering Body Of Knowledge) and Its Evaluation

Mikio Aoyama[1], Takako Nakatani[2], Shinobu Saito[3], Mikio Suzuki[4],
Kazuaki Fujita[5], Hiroaki Nakazaki[6], and Ritsuo Suzuki[7]
Nanzan University, 27 Seirei, Seto 489-0863, Japan, *mikio.aoyama@nifty.com*[1]
University of Tsukuba, 3-29-1 Otsuka, Bunkyo-ku, Tokyo 112-0012, Japan, *nakatani@gssm.otsuka.tsukuba.ac.jp*[2]
NTT DATA CORPORATION, 3-3, Toyosu 3-chome, Koto-ku, Tokyo 135-6033, Japan, *saitousnb@nttdata.co.jp*[3]
TIS, 14-5, Kaigan 1-chome, Minato-ku, Tokyo 105-8624, Japan, *mickey@tis.co.jp*[4]
Hitachi Systems & Services, Ltd., 2-18-1 Konan, Minato-ku, Tokyo, 108-8250, Japan, *k-fujita@hitachi-system.co.jp*[5]
FUJITSU FIP CORPORATION, Time 24 Bldg., 2-4-32 Aomi, Koto-ku, Tokyo 135-8686, Japan, *nakazaki@fip.fujitsu.com*[6]
Japan Information Technology Services Association, 2-8-1 Yaesu, Chuo-ku, Tokyo 104-0028, Japan, *rsuzuki@jisa.or.jp*[7]

Abstract—**Requirements engineering has been extensively developed as a discipline. Many statistics on the software development indicate requirements process is the most influential to both success and failure of software development. However, practitioners are still difficult to learn and apply requirements engineering. As a guideline for practitioners to learn and apply requirements engineering, we developed REBOK (Requirements Engineering Body Of Knowledge). In the development, we found there is no common model of BOKs in software engineering. This article proposes the model and architecture of the body of knowledge of REBOK and its proof of the concept.**

Keywords-requirements engineering, body of knowledge, software engineering education.

I. Introduction

Requirements engineering is known as the key to success to software and systems development [23]. However, practitioners are claiming the difficulty of leaning and applying requirements engineering to solving their problems. Our survey found that organizational education and training of requirements engineer is missing [5-7, 35, 36, 38, 40, 59].

On the other hand, requirements engineering community has been accumulating large body of knowledge with large number of literatures including books on principles and practice [16, 37]. There are BOKs (Body Of Knowledge) and certification programs related to requirements engineering. They include SWEBOK (Software Engineering Body Of Knowledge)[1], BABOK (Business Analysis Body Of Knowledge)[26] and CPRE (Certified Professional for Requirements Engineering) syllabus by IREB (International Requirements Engineering Board)[27]. However, they cover certain limited areas of requirements engineering but are not intended to cover the diverse knowledge and skills [21].

What's missing is the guide map to navigate practitioners and stakeholders involving in the requirements engineering.

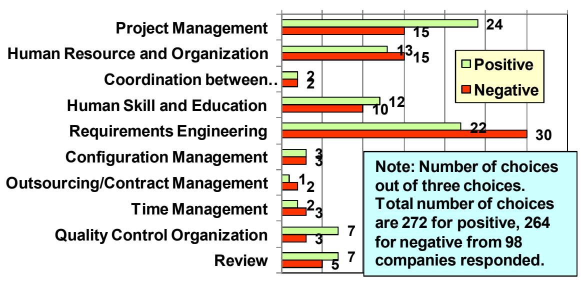
As a foundation of the guide map, we have been working for the development of REBOK (Requirements Engineering Body Of Knowledge) based on the intensive reviews of related BOKs aforementioned. This article proposes a model and architecture of REBOK. We also discuss the model of actors (i.e. stakeholders) and their learning objective levels involving in the requirements engineering.

Throughout this article, we intend to discuss on the body of knowledge on requirements engineering by asking questions: Who need it? How it should be? What it should include? How much we need to learn?

The structure of this article is as follows: Chapter II discusses the background and motivation. Chapter III review the related BOKs (Body Of Knowledge) followed by challenges of the development of REBOK in Chapter IV. Based on the approaches to the development of REBOK in Chapter V, we explain the knowledge model and BOK architecture of REBOK in Chapter VI. Chapter VII discusses learning objective model of REBOK. Chapter VIII and IX respectively discuss the evaluation of REBOK by surveys of opinions from practitioners and benchmark of REBOK with related BOKs. Finally, Chapter X illustrates future roadmap followed by conclusions in Chapter XI.

II. Background and Motivations

JISA (Japan Information Technology Services Association), the largest association of software industry in Japan, has been conducting the annual survey on the practical status of software development in Japan. Ninety eight major software companies responded. According to the survey illustrated in Figure 1, we found that requirements engineering is the most influential factor to both positive (better quality) and negative (poor quality) to software products. This result is consistent with other survey such as the Chaos Report.



Note: Number of choices out of three choices. Total number of choices are 272 for positive, 264 for negative from 98 companies responded.

Figure 1. Factors Influencing on Software Quality from JISA Survey.

1530-1362/10 $26.00 © 2010 IEEE
DOI 10.1109/APSEC.2010.16

In 2006, JISA initiated the RE WG (Requirements Engineering Working Group) to promote requirements engineering into practice. More than 100 people worked together for the RE WG. Most of them are practitioners working for either software developing companies or user companies. The RE WG published three technical reports annually while its three years of activities [5, 6, 7]. Through the discussions in RE WG, we found strong motivation to develop a common body of knowledge of requirements engineering. Thus, REBOK emerged in our mind.

On the other hand, as illustrated in Figure 2, we found strong necessity to develop organizational education and training program for requirements engineering professionals from some 100 responses at the requirements engineering symposium held in Tokyo late 2009. Some 78% participants, most of them are practitioners, say organizational education and training program for requirements engineering professionals are somewhat unsuccessful. This is also a strong motivation to develop REBOK.

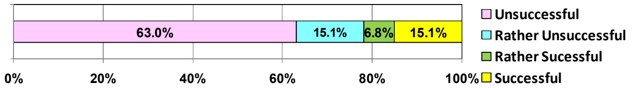


Figure 2. State of Organizational Development of Requirements Engineering Professionals.

III. Related Bodys Of Knowledge

We briefly review the following BOKs and certification programs closely related to requirements engineering as summarized in Table 1.

TABLE I. BOKs and Education Programs Related to Requirements Engineering

BOK/Syllabus	SWEBOK	BABOK	CPRE Syllabus
Latest Version	2004	Version 2 (‘09)	Version 2(’09)
Organization	IEEE CS	IIBA	IREB
Target Profession	Software Developer	Business Analyst	Persons Involved in RE
Scope	Software Engineering	Business Analysis	Requirements Engineering
UoK (Unit of Knowledge)	KU(Knowledge Unit)	KA(Knowledge Area)	EU (Educational Unit)
Number of UoK	7	7	9
Domain Dependency	No	Yes (Business)	No
Learning Objective Levels	Bloom’s Taxonomy	No	Specific to CPRE
Certification	CSDP, CSDA	CBAP	CPRE
Levels of Certification	Expert, Advanced	Expert, (Advanced)*	Basic,(Advanced, Expert)**

* Advanced is under development, **Advanced and expert are announced.

(1) SWEBOK (Software Engineering Body Of Knowledge) 2004 [1]: SWEBOK covers whole knowledge on software engineering. Chapter 2, software requirements, is devoted to the requirements engineering as a knowledge area of software engineering. The chapter consists of seven knowledge units in requirements engineering. We referred the software requirements of SWEBOK 2004 as the base of REBOK. A revision of SWEBOK is announced to publish in 2010 [52]. However, no major change on requirements engineering is announced.

(2) BABOK (Business Analysis Body Of Knowledge) [26]: BABOK is developed by IIBA (International Institute of Business Analysis) founded in 2003 by business analysis practitioners at Toronto, Canada. It published BABOK Version 2 in 2009, which consists seven knowledge areas in requirements engineering and business (enterprise) analysis. We reviewed the BABOK for the development of REBOK. IIBA has been conducting CBAP (Certified Business Analysis Professional) since 2006, and claimed to award the title to more than 100 professionals.

(3) CPRE (Certified Professional for Requirements Engineering) [27]: CPRE is a certification program developed by IREB (International Requirements Engineering Board) founded by a group of researchers in requirements engineering community in Europe. A syllabus for the foundation level of CPRE was published. Recently, advanced and expert levels of CPRE are announced. We referred the syllabus for foundation level of CPRE.

(4) Other BOKs related to certification programs: PMBOK (Project Management Body Of Knowledge) [44] is widely accepted for project management. Computing Curricula has been accepted [30, 41]. There are also local certification programs, such as ITSS (IT Skill Standard) by IPA, a government agency in Japan [28]. We discussed the ITSS during the development of REBOK but would not discuss here since its locality.

IV. Challenge

From an intensive survey of related BOKs abovementioned and literatures, we identified the following four major challenges in the development of REBOK.

(1) Lack of common model of BOKs: As indicated in Table I, the concept of knowledge is rather different among BOKs. SWEBOK uses a group of technical knowledge as a unit called KA (Knowledge Area) and KU (Knowledge Unit). Although BABOK and PMBOK use KA as a unit of the knowledge, KA means a procedure with a collection of tasks and is associated with its input and output. The difference of KA leads to the different architecture of BOKs. SWEBOK employs the hierarchical tree structure as the knowledge architecture, while BABOK employs workflow structure. Moreover, within the authors’ knowledge, there is no common architecture and its design techniques for BOKs.

(2) Lack of common knowledge area due to diversity of the knowledge, techniques and skills in requirements engineering: By its nature, requirements engineering encompasses diverse knowledge from theoretical principles, techniques and human skills [2, 8, 13, 14, 15, 24, 25, 29, 31, 32, 46, 47]. The scope of requirements engineering may vary from business analysis, system analysis and software analysis. Table II summarizes four *core* knowledge areas of requirements engineering

in SWEBOK, BABOK and IREB Syllabus. We can observe rather different concept of knowledge among three BOKs.

TABLE II. CORE KNOWLEDGES IN BOKS RELATED TO REQUIREMENTS ENGINEERING

KA/KU/EU	SWEBOK	BABOK	IREB Syllabus
Elicitation	KU w/ 2 Topics	KA w/ 4 Tasks	EU w/ 3 Units
Analysis	KU w/ 4 Topics	KA w /4 Tasks	No EU
Specification	KU w/ 3 Topics	No KA	3 EU w/ 16 Units
Validation	KU w/ 4 Topics	2KA w/ 4 Tasks	EU w/ 6 Units

(3) Lack of the common role model of requirements analyst: As pointed out in [22, 43, 49, 50], Requirements Engineer and BA (Business Analysis) are not established title comparing with, say, software architect. Therefore, the roles of Requirements Engineer and BA are not clearly defined.

(4) Lack of Practical Knowledge Body: Although excellent literatures presented on research directions [12, 42] and some literatures address practice of requirements engineering [3, 9, 17, 39, 48, 51, 55, 56, 58], little knowledge body has been developed for the practitioners of different levels of experiences. Therefore, many practitioners feel difficulties to learn and apply requirements engineering during their jobs and carrier development.

V. APPROACH: REQUIEMENTS ENGINEERING FOR REBOK

A. Engieering REBOK

Developing a BOK is a challenge. To make a BOK sound and consistent, we need an engineering approach to the development a BOK and to explain the development process and rationale of the BOK. Thus, we applied techniques of requirements engineering and software engineering to the development of REBOK.

Figure 3 illustrated the process of *concurrent* engineering REBOK we have gone through, which is actually elaborated while we are developing REBOK. Note it consists of two concurrent processes; one for the development of the model and another for the REBOK.

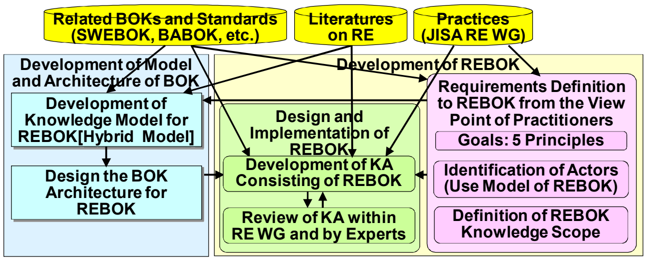

Related BOKs and Standards (SWEBOK, BABOK, etc.)
Literatures on RE
Practices (JISA RE WG)
Development of Model and Architecture of BOK
Development of Knowledge Model for REBOK[Hybrid Model]
Design the BOK Architecture for REBOK
Development of REBOK
Design and Implementation of REBOK
Development of KA Consisting of REBOK
Review of KA within RE WG and by Experts
Requirements Definition to REBOK from the View Point of Practitioners
Goals: 5 Principles
Identification of Actors (Use Model of REBOK)
Definition of REBOK Knowledge Scope

Figure 3. Concurrent Engineering Process for REBOK.

In June 2006, JISA initiated RE WG with practitioners from more than 30 companies and universities, and organized monthly workshop for three years since then. We studied both success and failure cases, and identified the root causes from various aspects including techniques, skills and management. We also invited research people on the state-of-the-art of requirements engineering, and conducted a survey on the related BOKs.

One of the most important concerns is whether REBOK is necessary while related BOKs including SWBOK and BABOK are already published. We must avoid developing duplicate BOKs. As the results of requirements elicitation process to REBOK, we defined the following goals of REBOK as five principles of REBOK.

B. Goals of REBOK: Five Principles

We defined the following five goals for the development of REBOK based on the intensive discussions in RE WG.

(1) REBOK should provide a set of common knowledge shared and used by practitioners in both users/customers and developers/vendors, and for both enterprise systems and embedded products, except for domain specific knowledge.

(2) REBOK should provide necessary knowledge to all the actors/stakeholders involving requirements engineering activity at different levels; the actors may include corporate management, end-user, project manager, and software developer.

(3) REBOK should provide a sound knowledge body in an appropriate manner over a clear architecture of three scopes of the knowledge; business/product, information system and software system.

(4) REBOK should be consistent with related BOKs including SWEBOK and BABOK in a certain way so that REBOK can be mapped to those BOKs.

(5) REBOK should be open to community so that everyone can use it and contribute it.

C. Requirements Elicitation for REBOK

We explored the following three sources to elaborate the *requirements* to REBOK; what's knowledge needed to practitioners.

(1) Practitioners: Through a series of workshops of JISA RE WG, we heard practitioners' opinions on the practical knowledge on requirements engineering and discussed the necessary knowledge to them.

(2) Related BOKs including SWEBOK and BABOK, and CPRE Syllabus

(3) Literature Survey: Books and papers on requirements engineering and its practice

D. Actors and their Roles in Reqirements Engineering

We identified three types of actors in the practice of requirements engineering as illustrated in Figure 4. Note that we distinguish *actors* and *stakeholders* in REBOK; actors mean participants in requirements engineering process while stakeholders mean participants in requirements.

(1) *RA (Requirements Analyst)*: Requirements Engineering Practitioners including Requirements Engineer, System Analyst [20, 53, 54], and BA (Business Analyst) [22, 26, 45] who apply and conduct requirements engineering to solving customers problems. As a generic name, here, we call RA (Requirements Analyst).

(2) *Users* including end-user, project manager, and software developer, who need to understand the outcome of requirements engineering.

(3) *Supporters* including corporate managers and CIO/CTO who need to understand the importance of requirements engineering and order to institutionalize requirements engineering as the corporate strategy.

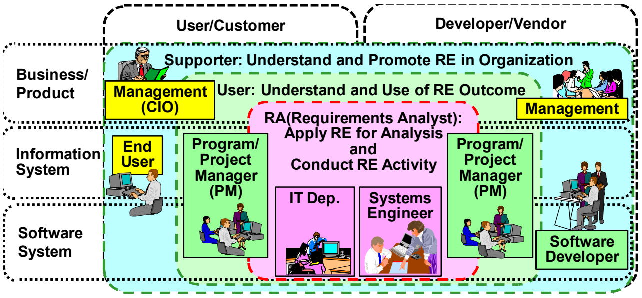

User/Customer
Developer/Vendor
Business/ Product
Information System
Software System
Supporter: Understand and Promote RE in Organization
Management (CIO)
User: Understand and Use of RE Outcome
Management
RA(Requirements Analyst): Apply RE for Analysis and Conduct RE Activity
End User
Program/ Project Manager (PM)
IT Dep.
Systems Engineer
Program/ Project Manager (PM)
Software Developer

Figure 4. Stakeholders in Requirements Engineering of REBOK.

E. Scopes and Associated Roles of Requirements Analyst in Requirements Engineering

Although “requirements engineering” is commonly used as an established discipline, practitioners apply the discipline under the different names of their role depending on the scope of requirements. We classified the scope of requirements into the following three layers, and identified the associated roles of “requirements analyst” as illustrated in Figure 5.

(1) Business and BA (Business Analyst)/ Product and Product Analyst: We assume the top level is either business for enterprise, or product for embedded or packaged solution. In this article, we focus on business. Although there is no standard definition of business analysis in organizations [49], IIBA provides a definition of business analysis [26], which seems to be a standard. BA (Business Analyst) is in charge of business analysis [22, 45]. Similarly, we use Product Analyst, but it might not be widely accepted definition yet. In many cases, marketing people takes the role.

(2) Information System and System Analyst: Information system means an integrated solution consisting both hardware and software. System analyst is rather broad but defined job title in U.S. [53], who in charge of transforming business and information requirements into specifications for information systems [20, 54].

(3) Software System and RA (Requirements Analyst) or Requirements Engineer: Software system is a home of requirements engineering. However, there is no accepted name of role who conducts requirements engineering [43]. Requirements Engineer is used in a certain community of software engineers. However, neither requirements engineer nor requirements analyst is commonly used. To specify the role, we temporally use Requirements Analyst and Requirements Engineer as the same role for the people conducting requirements analysis. Here, we suggest to initiating the discussions on the issue within our community in order to establish the role of conducting requirements engineering.

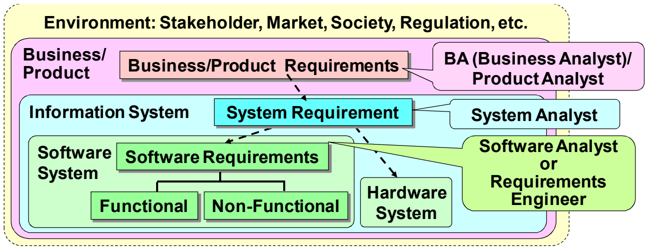

Environment: Stakeholder, Market, Society, Regulation, etc.
Business/ Product
Business/Product Requirements
BA (Business Analyst)/ Product Analyst
Information System
System Requirement
System Analyst
Software System
Software Requirements
Software Analyst or Requirements Engineer
Functional
Non-Functional
Hardware System

Figure 5. Scopes and Roles in Requirements Engineering of REBOK.

VI. REBOK Knowledge Model and Architecture

A. Proposed Unit of Knowledge of REBOK

As discussed before, there is no common model of knowledge for BOKs. Reviewing the related BOKs of SWEBOK and BABOK, we introduce the 3+1 levels of hierarchical knowledge units for REBOK; three layers of hierarchical knowledge units and a classification layer on top of the three layers of knowledge as follows.

(1) KA (Knowledge Area): Basic unit of knowledge. KA can be either *technical* type or *process* type defined below.

(2) KU (Knowledge Unit): A unit of knowledge element of a KA. In the process type KA, KU can be a task. A KU can be further decomposed into a set of KUs.

(3) T (Technique): A unit of knowledge, which can be independent to specific KA or KU, and used by multiple KAs or KUs.

We also introduce one virtual layer of Knowledge Category which is not substantial layer of knowledge but a classification layer of two categories of KAs, that is, REBOK Core and REBOK Extension.

Table III summarizes the hierarchical structure of the knowledge of REBOK and the related BOKs and syllabus.

TABLE III. Knowledge Model Of REBOK and Relatedx BOKs

REBOK	SWEBOK	BABOK	CPRE Syllabus
Knowledge Category	KA [10]*	-	-
KA	KU [7]	KA [7]	EU [9]
KU	Topic	Task[38]	Unit [36]
Technique	-	Technique [34]	-

* Requirements Engineering is a KA among 10 KAs of Software Engineering.

B. Review of Knowledge Models of Realted BOKs

Reviewing the related BOKs, we also found that concept of the knowledge is rather different among the BOKs. Looking at SWEBOK, BABOK, and PMBOK, we found that there are two types of knowledge in BOKs as follows.

(1) Technical Knowledge: An element of SWEBOK is a collection of technical knowledge including concept, principles and techniques. Therefore, KAs and subsequent KUs in SWEBOK are structured in a tree.

(2) Process Knowledge: An element of BABOK is a unit of work, which consists of a set of tasks. Both BABOK

and PMBOK are composed by a set of process KAs. Therefore, each KA is defined by its input, output and a set of tasks, and a set of stakeholders involving in the KA. Seven KAs of BABOK comprise 38 tasks. By the very nature of the process model, seven KAs of BABOK are structured with a fixed workflow and configure a process. Therefore, BABOK does not explicitly include "Requirements Engineering Process" as a KA. However, as exceptions, BABOK defines a set of *techniques* and a set of *competencies.* A set of techniques include 34 technical knowledge used across the KAs. Competencies include six basic knowledge and skills.

C. Proposed Hybrid Knowledge Model for REBOK

Considering the nature of requirements engineering, the knowledge types of REBOK should embrace both technical and process. Therefore, we introduce a *hybrid knowledge model* to REBOK by integrating the two types KAs and KUs as follows.

(1) Technical KA: Technical KAs and subsequent KUs of REBOK describe basic concept on requirements, principles on requirements, and techniques based on the principles. We introduce four technical KAs of "Requirements Engineering Fundamentals", "Requirements Engineering Process", "Requirements Planning and Management", and "Practical Consideration" into REBOK.

(2) Process KA: We regard four core KAs in SWEBOK including "Requirements Elicitation", "Requirements Analysis", "Requirements Specification" and "Requirements Validation" are process type. As discussed below, we introduce four process KAs of "Requirements Elicitation", "Requirements Analysis", "Requirements Specification", and "Requirements Verification, Validation and Evaluation" into REBOK. Process KAs are associated with input and output. A KA can be decomposed into a set of KUs which can configure a process within the KA.

D. Knowledge Category of REBOK Core

To identify knowledge areas for REBOK, we have extensively reviewed SWBOK, BABOK, and CPRE Syllabus. Since the knowledge models of SWEBOK and BABOK are rather different, we employed SWEBOK as the foundation for REBOK due to the following reason.

(1) A set of KAs of SWEBOK, found in many literatures on requirements engineering [16], conforms closely to the common understanding in requirements engineering and software engineering community and can be widely accepted.

(2) Conformity to Software Engineering: We expect REBOK can conform to SWEBOK in a certain way. Therefore, we should avoid unnecessary changes of SWEBOK at the level of seven out of eight KAs in REBOK core which are corresponding to seven KUs in SWEBOK.

As the result, we defined eight core KAs for REBOK as described in Table IV. We call those KAs *REBOK Core*.

TABLE IV. EIGHT KNOWLEDGE AREAS OF REBOK CORE

KA	Type	Definition
Requirements Engineering Fundamentals	Technical	Definition and essential properties on requirements.
Requirements Engineering Process	Technical	Concept and models of requirements engineering process.
Requirements Elicitation	Process	Sources and techniques for requirements elicitation
Requirements Analysis	Process	Techniques for analyzing requirements elicited
Requirements Specification	Process	Specification techniques for requirements analyzed
Requirements Verification, Validation and Evaluation	Process	Techniques validating requirements specification
Requirements Planning and Management	Technical	Properties, metrics and management techniques of requirements
Practical Consideration	Technical	Patterns and best practices for practicing requirements engineering

It appears that major difference is the addition of a KA of "Requirement Planning and Management". However, there are more differences at the level of KUs, which we re-organized topics of SWEBOK.

E. Knowledge Category of REBOK Extension

Unlike conventional BOKs which are closed, we introduce an idea of open BOK to REBOK, which is extensible through REBOK Extension Knowledge Category, or simply, *REBOK Extension*. Knowledge Area in REBOK Extension is syntactically the same to a KA in REBOK Core, but can be extended depending on the scope. At this moment, we assume two KAs in REBOK Extemsion, summarized in Table V, as follows.

(1) Enterprise Analysis: Enterprise Analysis is a KA in BABOK specialized to business analysis. We assume the Enterprise Analysis of REBOK, a process type KA, is corresponding to the Enterprise Analysis in BABOK. Therefore, at this moment, we are not intending to create yet another KA, but refer to the Enterprise Analysis of BABOK. Note that enterprise analysis can be called business analysis.

(2) Product Analysis: Product Analysis is a new KA for the knowledge on analysis of market-led products, including package and embedded products. For those products, requirement engineering is rather different among the products/product-lines and domains [46, 47]. Therefore, we intended to provide basic knowledge on product/product-line requirements engineering.

TABLE V. EIGHT CORE KNOWLEDGE AREAS OF REBOK

Extended KA	Type	Definition
Enterprise Analysis	Process	Definition and analysis of essential properties of enterprise business.
Product Analysis	Process	Definition and analysis of essential properties of products for market.

F. Knowledge Architecture of REBOK

We call knowledge architecture of REBOK as a structure configured by KAs. Like software architecture, we employ the following two views to the knowledge architecture.

(1) Structural View: Structure of REBOK

Figure 6 illustrates the hierarchical tree structure of REBOK knowledge architecture, which is considered as an extension of SWEBOK. The concept of hierarchical knowledge architecture is missing in BABOK and PMBOK since those BOKs are configured on the process.

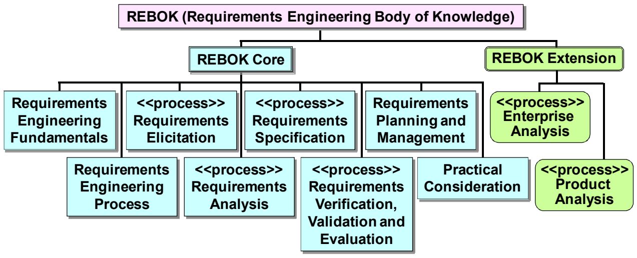

REBOK (Requirements Engineering Body of Knowledge)
REBOK Core
REBOK Extension
Requirements Engineering Fundamentals
<<process>> Requirements Elicitation
<<process>> Requirements Specification
Requirements Planning and Management
<<process>> Enterprise Analysis
Requirements Engineering Process
<<process>> Requirements Analysis
<<process>> Requirements Verification, Validation and Evaluation
Practical Consideration
<<process>> Product Analysis

Figure 6. Tree Strucuture of REBOK

(2) Behavioral View: Process of REBOK

Figure 7 illustrates the flow among six process KAs; four KAs in REBOK Core and two KAs in REBOK Extension. This process is not the same to the process of BABOK, but reflects the iterative process of requirements engineering presented in requirements engineering literatures [14, 31].

We do not assume a specific software development life-cycle process. However, we do discuss some aspects of the style of software development life-cycle process such as agile development in "Requirements Engineering Process".

As mentioned before, process KAs can be decomposed into a set of KUs, which can be configured to a process within the KA. Figure 8 illustrates the *internal* process of "Requirements Specification" with three KUs.

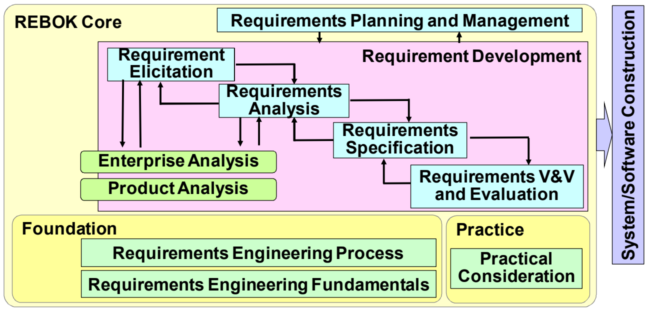

REBOK Core
Requirements Planning and Management
Requirement Development
Requirement Elicitation
Requirements Analysis
Requirements Specification
Requirements V&V and Evaluation
Enterprise Analysis
Product Analysis
Foundation
Requirements Engineering Process
Requirements Engineering Fundamentals
Practice
Practical Consideration
System/Software Construction

Figure 7. Process of REBOK

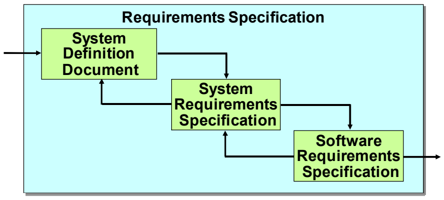

Requirements Specification
System Definition Document
System Requirements Specification
Software Requirements Specification

Figure 8. Process within Requirements Specification KA

G. Knowledge Units of REBOK

As of September 3rd, 2010, we defined 43 KUs in eight KAs of *Cosmos Version of REBOK Core*. As discussed later, we believe REBOK Core covers basic knowledge of requirements engineering.

Since REBOK is based on the SWEBOK, some of the KUs represent the same name of the Topics in SWEBOK. However, we reorganized some KUs (i.e. Topics in SWEBOK) across KAs. For example, "Iterative Nature of Requirements Process" is relocated from "Practical Consideration" to "Requirements Engineering Process".

KUs for *REBOK Extension* are under discussions.

Appendix A shows an example of KU description of "Software Requirements Specification", which is defined by considering the Specification Standard IEEE Std. 830-1998 and IEEE Std. 1233-1998. Note that the figure shows the description items only.

H. Practical Consideration KA in REBOK

We created two new KUs in the KA of "Practical Consideration": "RE Patterns" and "RE Best Practices".

We found two types of practical knowledge, patterns [6, 18, 57] and best practices [5, 51, 58]. We refer patterns as ones like design patterns, which is structurally defined in a way of pattern language. Literatures present a set of patterns [18, 57]. On the other hand, some books present a set of practical tips under the name of best practices, which is described in more informal ways [51, 58].

Those two types of knowledge are useful if the context of the practice is appropriate. Therefore, we introduce those patterns and best practices respectively into "RE Patterns" and "RE Best Practices" in REBOK. However, since they depend on the context of problems, it's not appropriate to include those patterns and best practices into REBOK Core. Rather, we provide references to the patterns and best practices in "RE Patterns" and "RE Best Practices" so that practitioners can find them from REBOK.

We, JISA RE WG, also conducted a series of workshops from 2007 to 2008 in order to collect patterns and best practices from the practice of requirements engineering in Japanese companies. We documented 35 patterns out of some 100 practices in the technical report [6].

VII. REBOK LEARNING OBJECTIVE LEVEL MODEL

A. Learning Objective Levels Based on Bloom's Taxonomy

We intended to use REBOK beyond professionals so that all the stakeholders can understand requirements engineering at certain level. Therefore, it's necessary to define learning objective levels to KAs. Even within the professionals such as Requirements Engineers and Business Analyst, it's necessary to specify a set of different objective levels of Basic, Advanced, and Expert, so that people can learn requirements engineering step by step with REBOK.

To define the educational and learning objectives, Bloom's taxonomy is a common framework and is widely used to evaluate curriculums [10, 11, 19, 33, 34]. The original taxonomy defines six levels of cognitive domains; Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation.

The taxonomy was revised and extended to two dimensions by adding a "Knowledge Dimension" which measures the types of knowledge elements [4]. The six cognitive domains are renamed to: Remember, Understand, Apply, Analyze, Evaluate and Create, all in verbs. The knowledge dimension is categorized to: Factual, Conceptual, Procedural, and Meta-Cognitive.

Among the related BOKs and a syllabus, SWEBOK and CPRE Syllabus define certain learning objective levels, but BABOK doesn't.

SWEBOK introduced Bloom's original taxonomy and defined either of three levels of Comprehension, Application, and Analysis, to each KU in requirements engineering [1].

CPRE Syllabus employs two levels of objectives; L1 (Knowing) and L2 (Mastering) [27]. Since CPRE Syllabus is intended to foundation level, L2 can be interpreted as Comprehension or Understand in the Bloom's taxonomy.

B. *REBOK Learning Objective Level Model*

Unlike the related BOKs and syllabus which are intended to only one class, i.e. level, of objective, REBOK need to classify the levels of objectives to three classes of actors, Analyst, User, and Supporter, involving in requirements engineering. Therefore, we extend the idea of Bloom's taxonomy to classify the necessary levels of knowledge to the three classes of actors. Table VI and VII illustrates our idea of classification of knowledge objective levels in REBOK, and its mapping to Bloom's taxonomy.

We set five levels of knowledge objective levels as shown in Table VI, which also reflects the idea of CMM(I) maturity level. Therefore, we can regard Table IV defines five maturity levels of professionals involving in the requirements engineering. As shown in Table VII, we define the mapping between the taxonomy of REBOK and those of Bloom's taxonomy of original [10] and revised [4], as follows.

(1) Level 1 (Know): Remember in Bloom's taxonomy revised is precise to indicate simple remembering. However, we use "Know" to indicate that people have certain knowledge on requirements engineering.

(2) Level 2 (Understand): This is the same to Bloom's taxonomy revised. Understand means the actor can understand the techniques of requirements engineering, and use the outcome such as requirements specification.

(3) Level 3 (Apply): This is also the same to the Bloom's taxonomy of original and revised. Apply means that actors can apply the techniques to requirements engineering, and can do certain tasks.

(4) Level 4 (Analyze and Evaluate): In the Bloom's taxonomy revised, Analyze includes differentiating, organizing and attributing, which are essential capabilities of analyst. Furthermore, we combined Analyze and Evaluate into one level, since analyze is essentially associated to evaluate in solving problems. Therefore, this level indicates core capability of analyst. Including Evaluate in level 4 also matches to level 4 of CMM, which requires quantitative evaluation.

(5) Level 5 (Create): In the Bloom's taxonomy revised, Create includes generating, planning and producing, which are capabilities of expert analyst, who is responsible to not simple solving problems, but also creating original idea, innovative solution, and conducting teams in problem solution.

Based on the discussions, Figure 9 illustrates *possible* two dimensional taxonomy of REBOK based on the Bloom's taxonomy revised [4] with the mapping in Table VII. Note that the two dimensional taxonomy is still under discussions.

TABLE VI. ROLE AND KNOWLEDGE LEVEL MODEL OF REBOK

Knowledge Objective Levels		Actors and their Roles		
		Supporter	User	Analyst
1	Know(Remember)	Manager	End User	-
2	Understand	-	PM, Developer	-
3	Apply	-	-	Basic
4	Analyze and Evaluate	-	-	Advanced
5	Create	-	-	Expert

TABLE VII. TAXONOMY MAPPIYPING

	REBOK Taxonomy	Bloom's Taxonomy Original [10]	Bloom's Taxonomy Revised [4]
1	Know	Knowledge	Remember
2	Understand	Comprehension	Understand
3	Apply	Apply	Apply
4	Analyze and Evaluate	Analyze	Analysis
		Evaluate	Evaluate
5	Create	Create	Create

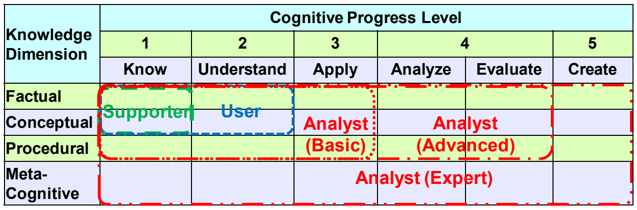


Figure 9. Knowledge Taxonomy of REBOK based on the Bloom's Taxonomy Revised.

VIII. RESPONSE FROM THE PRACTITIOERS

On October 2, 2009, we, JISA RE WG, organized a symposium to publicize the requirements engineering and REBOK. We conducted a survey to the participants from industry and received some 100 responses. From the responses, we found strong support to REBOK as shown in the following statistics.

(1) Strong support to REBOK: As shown in Figure 10, more than 90% participants favor the idea of REBOK, including strong favor of more than 50% participants.

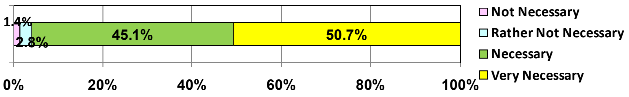


Figure 10. Necessity of REBOK.

(2) Disseminate the RE beyond requirements engineering professionals: We assume that knowledge on requirements engineering is necessary not only for the requirements engineering professionals but also for everybody involving in software development including end-user, project manager, and even corporate management, to certain degree. As illustrated in Figure 11, more than 95% participants support our view. They feel lack of the understanding at the corporate management is one of the major obstacles to promoting the practice of requirements engineering.

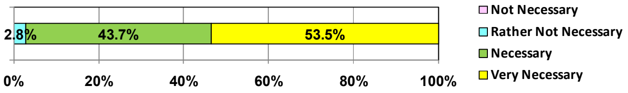


Figure 11. Necessityof the Undestanding of RE beyond Professionals.

(3) Multiple levels of learning objectives: The idea of multiple levels of learning objectives, say, Basic, Advanced, and Expert are also favored by more than 80% participants as illustrated in Figure 12.

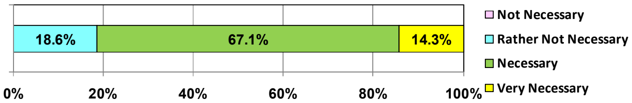


Figure 12. Necessityof the Multiple Lerning Levels of REBOK.

(4) Awareness on the related bodies of knowledge: To understand the awareness on the body of knowledge, we asked three related BOKs including SWEBOK, BABOK, and ITSS [28]. We found BABOK quickly gained the attentions of industry and became similarly popular to SWEBOK, as illustrated in Figure 13.

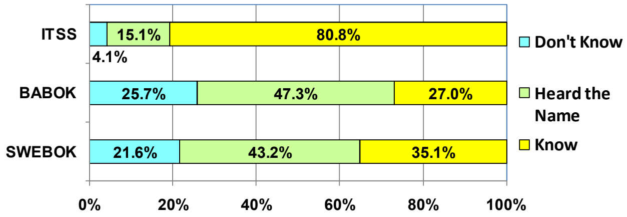


Figure 13. Status of Recognition of Related BOKs.

IX. Evaluation and Discussions

We developed a model and architecture, and associated knowledge and learning objective levels for actors involving in requirements engineering. Although the body of knowledge in REBOK is still under development, it's necessary to prove the proposed concept.

Here we, evaluate the model and architecture of REBOK in comparison with related BOKs and syllabus.

A. Comparison with Related BOKs and Syllabu

Table VIII summarizes the knowledge areas and levels of REBOK, SWEBOK, BABOK and IREB Syllabus. From Table VIII, we observe the following characteristics of BOKs and the syllabus.

(1) SWEBOK: Good coverage of KAs except for management. The major problems are lack of depth of knowledge and focus on software engineer. From practitioner's view, SWEBOK can be good entry point for software engineering as intended, but is limited due to the lack of the depth of knowledge.

(2) BABOK: Providing a set of practical procedures and a deep knowledge on enterprise analysis. However, the model and architecture is rather different from what requirements engineering community developed, which can be an obstacle to learn. Another problem is a narrow focus on enterprise domain, which also causes a limitation in the use of the BOK. These characteristics indicate that BABOK can be a good compliment to REBOK.

(3) IREB Syllabus: The foundation level provides a good entry pint to learn requirements engineering. However, since core capability such as analysis is missing, the practitioners may feel some gap between what they need and what the syllabus provides. It might be added to the advanced and expert levels planned in the future.

TABLE VIII. Comparison of REBOK with Related BOKs and IREB Syllabus

BOKs and Syllabus		REBOK	SWEBOK	BABOK	IREB
Model		Hybrid	Technique	Process	Technique
Architecture		Tree and Process	Tree	Process	Can be Tree (Not Specified)
Actors		Analyst, User, Supporter	Software Engineer	Business Analyst	Requirements Engineer
KA	Fundamentals	6 KU	6 Topic	6 Competency	2EU, 3 Unit
	Process	5 KU	4 Topic	-	-
	Elicitation	7 KU	2 Topic	4 Task	1EU, 3Unit
	Analysis	5 KU	4 Topic	6 Task (incl. 2 for Validation)	-*
	Specification	4KU	3 Topic	-	3EU, 16Unit
	V&V, Evaluation	6 KU	4 Topic	6 Task	1EU, 6Unit
	RE Management	8 KU	-	2KA, 11Task	1EU, 6Unit
	Practice	2 KU	5 Topic	-	
	Tool Support	Techniques	-	-	1EU, 3Unit
	Enterprise Analysis	Under Dev.	-	5 Task	-
	Product Analysis	Under Dev.	-	-	-
Technique		Technique	-	34 Technique	-
Objective Levels		5	3 out of 5	-	2
Certification Levels	Expert	3 Levels**	CSDP	CBAP	Expert*^{4}
	Advanced	1 Level**	CSDA	*^{3}	Advanced*^{4}
	Basic	1 Level**	-	-	Foundation

* One EU in Specification covers Model-based Documentation, **Not for Certification, *^{3} Entry level is under planning, *^{4} Higher levels are under planning.

From the compassion, we believe REBOK can provide a good model and architecture to the BOK on requirements engineering with the following characteristics:

(a) Clear and extensible BOK architecture for diverse knowledge of requirements engineering: Within our knowledge, REBOK is the first BOK to clearly define the model and architecture of BOK. The hybrid architecture proposed embraces the diversity of the knowledge on requirements engineering from business to software.

(b) Clear model of actors: REBOK defines the actors involving in the requirements engineering, and extended the scope of the BOK from analyst to end-users and managers.

(c) Practical and balanced coverage on requirements engineering: REBOK is collaboratively developed by both expert practitioners and researchers with a wide survey of literatures on requirements engineering. It covers the knowledge on both technical and process in a balanced way.

(d) Learning level: REBOK defines the clear learning objective level based on the Bloom's taxonomy revised [4] which is widely accepted. This taxonomy can guide practitioners and educators to plan learning scenario.

B. *Contribution of the Model and Architecture of REBOK*

We believe the major contributions of REBOK include the model and architecture of BOK, and associated role model of actors and learning objective levels. These aspects of BOK are not discussed yet in requirements engineering community. Since the requirements engineering is unique in its diversity of knowledge, it's important to discuss how to cope with the diversity in the education and training in both academic and professional education programs [38, 59]. Therefore, with the learning objective levels, REBOK can provide a guideline to the development of the education programs.

Another important contribution of REBOK is, we believe, the extension of the scope of requirements engineering. Many practitioners argue that importance of requirements engineering largely ignored at out of the community. This is the major obstacle to the penetration of the requirements engineering into practice. Therefore, REBOK is intended to extend the scope of requirements engineering to the managers and end-users. This leads to the five levels of learning objectives so that actors can set an appropriate goal in the learning of requirements engineering. This is particularly important to extend the value of requirements engineering into our society.

X. FUTURE ROADMAP

Open and *incremental* are our policy to the development of REBOK. Open means two folds. First, development process is open to community. Open collaboration with our community is indispensable to develop and share the large BOK. We would like to collaborate with requirements engineering community to develop REBOK. Second, we designed the architecture of REBOK extensible. To avoid unnecessary duplication, we do not intend to make duplicate BOK. Therefore, we refer other BOKs and literatures, if appropriate. At this moment, enterprise analysis KA of REBOK is an extension so that it may refer the enterprise analysis of BABOK.

We are working together to develop REBOK. We plan to release Cosmos version at the end of September, 2010 for public use in Japan. And, a round table discussion on REBOK is scheduled at the International Requirements Engineering Conference to be held in Sydney in October 2010 (http://www.re10.org).

JISA has been hosted the development of REBOK and committed to support for future development. So, we are planning to release revised versions incrementally.

XI. CONCLUSIONS

This article proposed the model of actors and associated knowledge on requirements engineering, and the hybrid architecture of REBOK based on the extensive review of literatures on requirements engineering and related BOKs including SWEBOK, BABOK and IREB syllabus.

Within our knowledge, there is no common model and architecture of BOKs in software engineering and requirements engineering. Therefore, this article contributes to first propose a common model and architecture of BOKs in requirements engineering.

The model and architecture of REBOK are intended to embrace the diversity of knowledge on requirements engineering, and help practitioners to learn and practice requirements engineering. Therefore, we, a number of expert practitioners and researchers, worked together to develop REBOK from practitioners' point of view.

As the next step, we will initiate the discussions on REBOK in global requirements engineering community, and will work together with experts from all over the world in order to develop full-fledged REBOK. We welcome your contributions.

ACKNOWLEDGMENT

The authors thank to JISA (Japan Information Technology Services Association) for the continuous support to the development of REBOK.

APPENDIX A

Description of Software Requirements Specification KU

5.3 Software Requirements Specification
5.3.1 Definition
5.3.2 Purpose
5.3.3 Stakeholder
5.3.4 Component
(1) Introduction
1) Purpose
2) Scope
3) Overview
(2) Overall Description
1) Product Perspective
a) System Interface, b) User Interface, c) Hardware Interface
d) Software Interface, e) Communication Interface
f) Operation, g) Site Adaptation Requirements
2) Product Functions
3) User Characteristics
4) Constraints
5) Assumptions and Dependencies
(3) Specific Requirements
1) System Characteristic
2) External Interface Requirements
3) Functional Requirements
4) Non-Functional Requirements
(4) Development Plan and Estimation
(5) Performance Estimation
5.3.5 Techniques
5.3.6 Input
5.3.7 References
5.3.8 Output
5.3.9 Related Knowledge Units
5.3.10 References

REFERENCES

[1] A. Abran, et al. (eds.), *Guide to the Software Engineering Body of Knowledge, 2004 Version*, IEEE Computer Society, http://www.swebok.org/ (Retrieved Sep. 2010).

[2] I. Alexander and N. Maiden, *Scenarios, Stories, Use Cases*, Wiley, 2004.

[3] I. Alexander and L. Beus-Dukic, *Discovering Requirements*, Wiley, 2009.

[4] L. W. Anderson, et al. (eds.), *A Taxonomy for Learning, Teachning and Assessing: A Revison of Bloom's Taxonomy of Educational Objectives*, Longman, 2001.

[5] M. Aoyama, et al., *Requiremnts Engineering Best Pracices*, JISA Tech. Rep., No. 18-J008, Mar. 2007 (In Japanese).

[6] M. Aoyama, et al., *Requirements Engineering Patters from the Bestpractices*, JISA Tech. Rep., No. 19-J004, Mar. 2008 (In Japanses).

[7] M. Aoyama, et al., *REBOK (Requirements Engineering Body Of Knowledge)*, JISA Tech Rep., No. 20-J007, Mar. 2009 (In Japanese).

[8] A. Aurum and C. Wohlin (eds.), *Engineering and Managing Software Requirements*, Springer, 2005.

[9] B. Berenbach, et al., *Software & Systems Requirements Engineering: In Practice*, McGraw Hill, 2009.

[10] B. S. Bloom (ed.), *The Taxonomy of Educational Objectives: Handbook 1: Cognitive Domain*, Longman, 1956.

[11] P. Bourque, et al., Bloom's Taxonomy Levels for Three Software Engineer Profiles, *Proc. STEP 2004*, IEEE Computer Society, Sep. 2004, pp. 123-129.

[12] B. H. C. Cheng and J. M. Atlee, Research Directions in Requirements Engineering, *Proc. ICSE 2007, Future of Software Engineering*, IEEE Computer Society, May 2007, pp. 285-303.

[13] L. Chung, et al., *Non-Functional Requirements in Software Engineering*, Kluwar Academic, 1999.

[14] J. Cleland-Huang, Software Requirements, R. H. Thayer, et al. (eds.), *Software Engineering*, 3rd ed., IEEE Computer Society, 2005, pp. 123-139.

[15] A. M. Davis, *Software Requirements*, Prentice-Hall, 1990.

[16] A. M. Davis, *Requirements Bibliography*, http://www.reqbib.com/.

[17] S. Faily, Towards Requirements Engineering Practice for Professional End User Developers: A Case Study, *Proc REET 2008*, IEEE Computer Society, Sep. 2008, pp. 38-44.

[18] M. Fowler, *Analysis Patterns*, Addsion-Wesley, 1997.

[19] U. Fuller, et al., Developing a Computer Science-Specific Learning Taxonomy, *ACM SIGCSE Bulletin*, Vol. 39, No. 4, Dec. 2007, pp. 152-170.

[20] J. O. Grady, *System Requirements Analysis*, Academic Press, 2006.

[21] M. Glinz and R. J. Wieringa, Stakeholderd in Requirements Engineering, IEEE Software, Vol. 24, No. 2, Mar./Apr. 2007, pp. 18-20.

[22] K. B. Hass, *Professionalizing Business Analysis*, Management Concepts Inc., 2007.

[23] H. F. Hofmann and F. Lehner, Requirements Engineering as a Success Factor in Software Projects, *IEEE Software*, Vol. 18, No. 4, Jul./Aug. 2001, pp. 58-66.

[24] C. Hood, et al., *Requirements Management*, Springer, 2007.

[25] E. Hull, et al., *Requirements Engineering*, 2nd ed., Springer, 2004.

[26] IIBA, *A Guide to the Business Analysis Body of Knowledge (BABOK Guide)*, Version 2.0, IIBA, 2009.

[27] IREB, *Syllabus: IREB Certified Professional for Requirements Engineering -Foundation Level-*, Ver. 2.0, Oct. 2009, http://certified-re.de/en/syllabi.html (Retrieved Sep. 2010).

[28] IPA , *IT Skill Standard, V3 2008*, 2008, http://www.ipa.go.jp/jinzai/itss/download_V3_2008.html (Retrieved Sep. 2010).

[29] M. Jackson, *Problem Frames*, Addison-Wesley,2001.

[30] Joint Task Force for Computing Curricula 2005, *Computing Curricula 2005: The Overview Report*, Sep. 2005, http:// www.acm.org/education/curricula-recommendations (Retrieved Sep. 2010).

[31] G. Kotonya and I. Sommerville, *Requirements Engineering*, Wiley, 1998.

[32] A. van Lamsweerde, *Requirements Engineering*, Wiley, 2009.

[33] T. C. Lethbridge, What Knowledge is Important to a Software Professionals?, *IEEE Computer*, Vol. 33, No. 5, May 2000, pp. 44-50.

[34] T. C. Lethbridge, et al., SE 2004: Recommendations for Undergraduates Software Engineering Curricula, *IEEE Software*, Vol. 23, No. 6, Nov./Dec. 2006, pp. 19-25.

[35] T. C. Lethbridge, et al., Improving Software Practice through Education: Chalenges and Future Trends, *Proc. ICSE 2007, Future of Software Engineering*, IEEE Computer Society, May 2007, pp. 12-28.

[36] L. Macaulay and J. Mylopoulos, Requirements Engineering; An Educational Dilemma, *J. of Automated Software Eng.*, Vol. 2, No. 4, Dec. 1995, pp. 343-351.

[37] N. Mead, *Requirements Engineering Annotated Bibliography*, https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/requirements/231-BSI.html (Retrieved Sep. 2010).

[38] T. Nakatani, et al., Instructional Design of a Requirements Engineering Education Course for Professional Engineers, G. A. Tsihrintzis, et al. (eds.), *Multimedia Services in Intelligent Environments*, Vol. 3, Springer, 2010.

[39] C. J. Neil and P. A. Laplante, Requirements Engineering: The State of the Practice, *IEEE Software*, Vol. 20, No. 6, Nov./Dec. 2003, pp. 40-45.

[40] U. Nikula, et al., *A State-of-the-Practice Survey on Requirements Engineering in Small- and Medium-Sized Enterprises*, Telecom Business Research Center Lappeenranta Research Rep., 2000, http://www.tbrc.fi/pubfilet/TBRC_500000146.pdf (Retrieved Sep. 2010).

[41] D. Notkin, et al., *An Assessment of Software Engineering Body of Knowledge Efforts*, May 2000, http://www.cs.washington.edu/homes/notkin/bok_assessment.pdf (Retrieved Sep. 2010).

[42] B. Nuseibeh and S. M Easterbrook, Requirements Engineering: A Roadmap, *Proc. ICSE 2000, The Future of Software Engineering*, ACM, May 2000, pp. 35-46.

[43] B. Paech, What Is a Requirements Engineer?, *IEEE Software*, Vol. 25, No. 4, Jul./Aug. 2008, pp. 16-17.

[44] PMI, *A Guide to Project Management Body of Knowledge*, 4th ed., PMI, 2008.

[45] H. Podeswa, *The Business Analyst's Handbook*, Course Technology, 2009.

[46] K. Pohl, et al., *Software Product Line Engineering: Foundations, Principles and Techniques*, Springer, 2005.

[47] K. Pohl, *Requirements Engineering*, Springer, 2010.

[48] S. Robertson and J. Robertson, *Mastering the Requirements Process*, 2nd ed., Addison-Wesley, 2006.

[49] J. Rubens, Business Analysis and Requirements Engineering: The Same, Only Different?, *Req. Eng. J.*, Vol. 12, No. 2, Apr. 2007, pp. 121-123.

[50] K. Schreiner, The Bridge and Beyond: Business Analysis Extends Its Role and Reaches, *IEEE IT Professional*, Vol. 9, No. 6, Nov./Dec. 2007, pp. 50-53.

[51] I. Sommerville and P. Sawyer, *Requirements Engineering: A Good Practice Guide*, Wiley, 1997.

[52] SWEBOK, *SWEBOK Guide Set for 2010 Refresh*, http://www.computer.org/portal/web/swebok (Retrieved Sep. 2010).

[53] U.S. Bureau of Labor Statistics, *Computer Systems Analysts*, http://www.bls.gov/oco/ocos287.htm (Retrieved Sep. 2010).

[54] J. L. Whitten and L. D. Bentley, *Systems Analysis and Design Methods*, 7th ed., McGraw Hill, 2007.

[55] K. E. Wiegers, *Software Requirements*, 2nd ed., Microsoft Press, 2003.

[56] K. E. Wiegers, *More About Software Requirements: Thorny Issues and Practical Advice*, Microsoft Press, 2006.

[57] S. Withall, *Software Requirement Patterns*, Microsoft Press, 2007.

[58] R. R. Young, *The Requirements Engineering Handbook*, Artech House, 2004.

[59] D. Zowghi, Requirements Engineering Education and Training: Key Challenges and Practical Solutions, *Proc. RE '09*, IEEE Computer Society, Sep. 2009, p. 358.