A Novel Approach for Integrated CBOK with Software Engineering Body of Knowledge Using Formal Mapping Technique

First A. Kenza Meridji

Abstract**—** To date in software engineering discipline literature mapping is done in a non formal way. Although software engineering is a new emerging engineering discipline but still lacks formal mapping methods. The most known body of knowledge is the SWEBOK that is considered as a body of knowledge for software engineering discipline that supports program curriculums and certifications. The GSwE2009 curriculum includes the core body of knowledge CBOK which provides guidelines for the education of students for a professional master's degree in software engineering. This paper suggests a novel approach to mapping using set theory operations. This mapping method will make formal mapping between two defined entities or sets. This research uses the software engineering body of knowledge (SWEBOK) and a core body of knowledge (CBOK) as object of study; and therefore the formal mapping is applied for both bodies of knowledge at all levels of structural decomposition; and consequently, the differences and similarities between both bodies of knowledge are depicted.

Keywords**—** Software Engineering, SWEBOK, CBOK, Set Theory

I. Introduction

Software engineering is defined by the Institute of Electrical & Electronics Engineers (IEEE) as: (1) "The application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software, i.e. the application of engineering to software. (2) "The study of approaches" as in (1) [4].

The need for software engineering is increasing due to the large scale software development project. While many universities teach software engineering programs at the undergraduate's level, as well as at a master level, the evolution of software engineering towards professional status is also determined by the development of a core body of knowledge, this is a key objective in all disciplines to be able to reach consensus by a profession, and this has been recognized as fundamental by the IEEE Computer Society.

This work was supported in part by the U.S. Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd–Fe–B"). Do not write "(Invited)" in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors' initials.

F. A. Kenza Meridji is now with the University of Petra, Collage of Information Technology, Department of Software Engineering, 11196 Amman, Jordan. (e-mail: kmeridji@uop.edu.jo).

Over the years few software engineering curriculums and body of knowledge have been developed.

To develop such an international consensus on a body of knowledge for software engineering, the IEEE Computer Society's Professional Practices Committee started a project in 1998 and published two versions of the guide to the software engineering body of knowledge (SWEBOK) in 2001 and 2004 [6]. In 2005 the 2004 version was also accepted as an international ISO Technical Report 19759 [9]. The current version 3 of the SWEBOK Guide was developed and reviewed by professionals, and released in 2014 [5].

In 2009 representives from academia, industry, government, and professional societies joined together to develop a graduate program for software engineering curriculum guidelines GSwE2009 that was published in 2009. Its objective is to make available guidelines for professionals as well as for Master's program in software engineering. The core content of this curriculum is known as the CBOK or Core Body of Knowledge [8].

This paper explores the differences between the base lined breakdown of SWEBOK Guide V3 and the CBOK (2009) in terms of breadth and depth breakdown topics. Basic set theory operations are used to perform a formal mapping between both breakdowns. This formal mapping method is systematic unlike the existing non formal mapping. Consequently, this will allow having an accurate result by minimizing mistakes.

The investigation of such differences can enhance the content of the SWEBOK and CBOK as well as enhancing academic curriculum and course material in software engineering based on these bodies of knowledge. Moreover this investigation can facilitate an improved comprehension of small differences or even different school of thought concerning how the discipline of software engineering is understood and practiced as a profession.

This paper is organized as follows: Section 2 presents reviews of the literature. Section 3 describes the mapping decomposition levels. Mapping methodology is introduces in section 4. Section 5 illustrates the application of mapping at the first level of breakdown. Section 6 presents the mapping at second level of breakdown. Results of all levels of decomposition are provided in section 7. A conclusion and future work are described in section 8.

II. Literature Review

This section describes the related inputs to this research study. The SWEBOK Guide embodies the first software engineering body of knowledge released in 2004 [6]. Also the most widely used body of knowledge for software engineering is the SWEBOK Guide; as well as the core body of knowledge (CBOK) of GSwE2009. Set theory operations are also documented in this literature as constituting the mathematical foundations for software engineering.

A. Guide to the software engineering body of knowledge (SWEBOK)

The Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), written under the auspices of the IEEE Computer Society's Professional Practices Committee, was initiated in 1998 to develop an international consensus in pursuing many objectives such as: documenting the software engineering discipline content; supporting consistent view of software engineering worldwide; making available the body of knowledge for software engineering; defining the scope of software engineering with respect to other disciplines; defining a foundation for developing curriculums as well as giving certification material to individual.

In 2004, the IEEE Computer Society and ISO published a guide the software engineering body of knowledge – the SWEBOK Guide (ISO 19759 2004) [9]. In 2014, version 3 of the SWEBOK is published [5].

The current baseline breakdown of topics of SWEBOK Guide V3 contains 15 Knowledge Areas KAs and is composed of three level structures. Among them eleven knowledge areas characterizing the practice of software engineering: software requirements (SR), software design (SD), software construction (SC), software testing (ST), software maintenance (SM), software configuration management (SCM), software engineering management (SEM), software engineering process (SEP), software engineering models and methods (SEMM), software quality(SQ) and software engineering professional practice (SEPP). An additional four Knowledge areas characterizing the educational requirements of software engineering: Software engineering economics (SEE), computing foundations (CF), mathematical foundations (MF) and engineering foundations (EF).

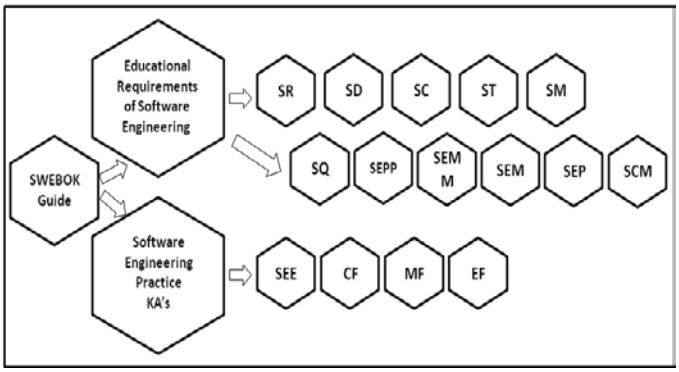

Fig. 1: SWEBOK Guide Knowledge Areas

B. Core body of knowledge CBOK

In 2007, various authors from academia, industry, government, and professional societies joined together to produce a curriculum for software engineering in a project named the Integrated Software and systems engineering curriculum (iSSEc).

As a result, the first product of the iSSEc project is the graduate software engineering 2009 (GSwE2009): curriculum guidelines for graduate degree programs in software engineering.

The primary goal of GSwE2009 is to provide guidelines for the education of students for a professional or primarily course-based master's degree in software engineering. This guideline notably identifies the core body of knowledge CBOK for graduate students of a master program. The CBOK therefore represents the core of the curriculum and is composed of three levels structure knowledge areas, subareas and topics. It includes the following knowledge areas: ethics and professional conduct (EPC), system engineering (SYE), requirements engineering (RE), software design, software construction (SC), testing (T), software maintenance (SM), configuration management (CM), software engineering management (SEM), software engineering process (SEP) and software quality (SQ). The CBOK also includes three knowledge areas introduced as preparatory knowledge for core body of knowledge areas present in CBOK such as mathematics fundamentals (MF), computing fundamentals (CF) and software engineering (SE).

The GSwE2009 build on the Software Engineering Institute SEI curriculum [18] in addition to the Guide of Software Engineering Body of Knowledge (SWEBOK) and Software Engineering: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (SE2004) [11].

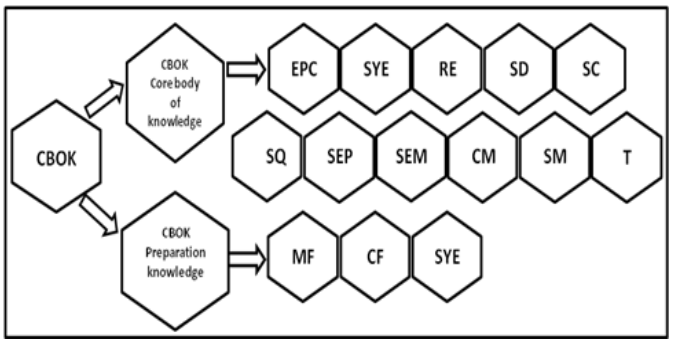

Fig. 2: CBOK Knowledge Areas

C. Literature work using non formal mapping methods

To date in the literature of software engineering the mapping is done in a non formal way leading to non precise results and a lack of traceability. Prior work has been done concerning such mapping for instance in the study conducted in a workshop at Software Technology and Practice Conference (STEP 2002) a preliminary mapping was undertaken between two bodies of knowledge the SWEBOK and the Software Engineering Education Body of Knowledge (SEEK) [12]. As well as in a work entitled "Software engineering principles: do

they meet engineering criteria" mapping the principles for software engineering to engineering criteria [2] and in "Software Engineering Principles: A Survey and an Analysis" mapping principles to principles criteria [7],

D. Set theory in mathematic

Set theory was invented by Georg Cantor in 1895. It is the branch of mathematics that studies sets. Set theory characterizes a base for modern mathematics and all formal description utilizes set theory. This mathematical specification can then be analyzed to prove the correctness and consistency of the domain under study. Basic concepts of set theory are: Sets elements can be viewed as a collection of objects, concepts or vocabularies which are called the members or elements of that set. Sets can be finite, infinite, empty set, singleton set. The empty set, or called also null set doesn't have any elements. A singleton or a singleton set has only one element. Many operations can be performed on sets. The basic set operations are union, intersection and difference.

E. Set theory in software engineering foundations

In 2007 Wang explored the theoretical and the organizational foundations of software engineering in a book entitled "Software engineering foundations" [3]. Among these theoretical foundations Wang described the mathematical foundations for software engineering discipline to comprehend the role of mathematics for the discipline such as classical mathematic: set theory.

F. Set theory in the bodies of knowledge

SWEBOK and CBOK guide dedicated two knowledge areas about mathematical foundations and mathematical fundamentals among the sections that describe the educational requirements of software engineering and preparation knowledge. These knowledge areas encompass the description of many elements and concepts used in mathematics such as set theory in their breakdown of topics. They are also designed to provide support for software engineers to be able to comprehend the logic used in programming. As well as it provides researchers some mathematics fundamentals that can be used by researchers working in the foundations of software engineering field.

The basics set operations and its attributes can be defined for both the SWEBOK and CBOK knowledge areas as follows:

The set of knowledge areas =

$$\{e|e \in S \vee e \in C\} \quad \text{... (1)}$$

$$S \cup C == \{e|e \in S \vee e \in C\} \quad \text{... (2)}$$

$$S \cap C == \{e|e \in S \wedge e \in C\} \quad \text{... (3)}$$

$$S - C == \{e|e \in S \wedge e \notin C\} \quad \text{... (4)}$$

Letters and symbols used in the above equations:

- The knowledge areas represented by element (e);
- SWEBOK guide as a main set represented by (S) and the CBOK as second main set represented by (C).
- The symbols ($\vee$, $\wedge$) mean or, and, respectively.
- The symbols ($\in$ and $\notin$) mean that the element belongs to and does not belong respectively.
- The symbols ($\cup, \cap, -$) mean union, intersection and difference.

III. Mapping levels: decomposition structure of the SWEBOK and CBOK guides

The SWEBOK and the CBOK guides adopt a three level decomposition and names each level consistently as: knowledge areas, subareas and topics - figure 3. The formal mapping approach will be performed at all levels of decomposition using the mapping methodology described in figure 4.

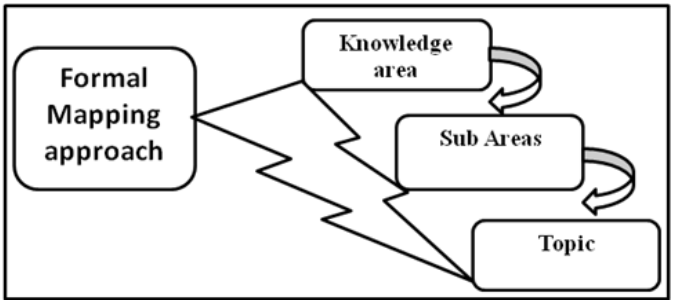

Fig. 3: Mapping levels decomposition

A. Formal mapping methodology

To tackle the issue of formal mapping in software engineering, set theory operations will be used. The approach designed for formal mapping by means of set theory and applying it to map between the SWEBOK and the CBOK. This consists of three phases methodology- see figure 4.

This methodology is composed of three phase's initiation mapping, practical mapping and results analysis.

1) Phase 1: Initiation phase

The initiation phase is composed of three steps and is described as follows:

Step 1- Sets definition: This step consist of describing the sets that will be the object of mapping for instance one set for the SWEBOK knowledge areas and the other set for the CBOK knowledge areas.

Step 2- Symbolize sets: In this step the output of the previous step are taken as input, that is, the two sets for both SWEBOK and CBOK. Each one of these set is composed of knowledge areas or elements. Then, each of these knowledge area or elements in both sets will get an acronym or a symbol.

Step 3- Sets associations: This step defines the visual links that exist between the different knowledge areas of both the SWEBOK and the CBOK at any level of mapping whether knowledge areas, subareas and the breakdowns.

2) Phase 2: Practical phase

The practical phase is composed of three steps and is described as follows:

Step 1: Application of union: The predicate notation of a union operation is defined and then applied to define the elements that can be represented whether by knowledge areas, subareas or topics which are described in the SWEBOK or in the CBOK.

Step 2: Application of intersection: This step defines the predicate notation of intersection operation then applied to illustrate the elements that can be represented whether by knowledge areas, subareas or breakdown which are described in both the SWEBOK and in the CBOK.

Step 3: Application of difference: The predicate notation of the difference operation is described then applied to identify the elements that can be represented whether by knowledge areas, subareas or breakdown which belong to SWEBOK and not to CBOK and vice versa.

3) Phase 3: Results analysis phase

The results analysis phase consists of collecting results after performing the initiation and the practical phase and then analysing them.

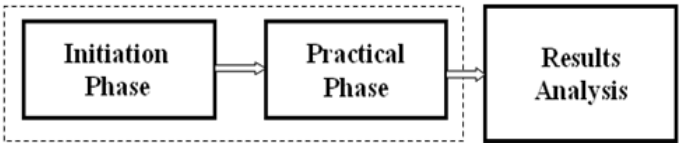

Fig. 4: Formal Mapping Methodology

B. Mapping between SWEBOK and CBOK at knowledge area level

This section introduces the two phase's methodology for the formal mapping at first level of breakdown using set theory: initiation mapping phase and practical mapping phase.

1) Phase 1: Initiation mapping phase

Initiation mapping phase for the SWEBOK and CBOK guides includes the following steps:

Step 1: Sets definition for SWEBOK and CBOK at first level of breakdown:

For a hierarchal mapping one can consider a SWEBOK and CBOK as a main two sets and each knowledge areas as subsets from the main sets in each guide. In this section, the set definition for the SWEBOK and CBOK guides and their knowledge areas as elements or (attributes) of the SWEBOK and CBOK sets are built based on the following equation:

The set of knowledge areas that belong to SWEBOK or to CBOK = $\{e|e \in S \vee e \in C\}$

SWEBOK (S)= {Software requirements (SR), Software design (SD), Software construction (SC), Software testing (ST), Software maintenance (SM), Software configuration management (SCM), Software engineering management (SEM), Software engineering process (SEP), Software engineering methods (SEMM), Software quality (SQ), Software engineering professional practice (SEPP), Software Engineering economics (SEE), Computing foundations (CM), Mathematical foundations (MF) and Engineering foundations (EF)}

CBOK (C) = {Ethics and professional conduct (EPC), System engineering (SYE), Requirements engineering (RE), Software design (SD), Software construction (SC), Testing (T), Software maintenance (SM), Configuration management (CM), Software engineering management (SEM), Software engineering process (SEP), Software quality (SQ), Mathematical fundamentals, Computing fundamentals, software engineering (SE)}

Step 2: Symbolize sets definition for SWEBOK and CBOK at first level of mapping:

This section describes both the SWEBOK and the CBOK sets using symbols. The following is *set S* containing knowledge areas from the SWEBOK Guide

S==
{SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF, MF,EF}

The following is *set C* containing knowledge areas from the CBOK.
C== {EPC, SE, RE, SD, SC.T, SM, CM, SEM, SEP, SQ}

Step 3: Sets associations at the first level of mapping:

Figure 5 describes the relations that exist between the SWEBOK Guide and the CBOK at knowledge areas level. The related links illustrates the common knowledge areas that are present in the SWEBOK as well as the CBOK. For instance, software requirements (SR) in SWEBOK is the same as in the CBOK but with a different name, requirements engineering (RE) and system engineering is not present in the SWEBOK.

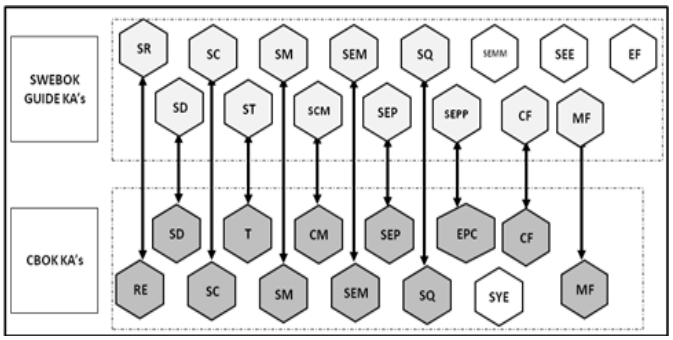

Fig. 5: A visual association between SWEBOK and CBOK at first level of breakdown

2) Phase 2: Practical mapping phase

This section describes an example of the mapping based on the set theory at the knowledge area level. The following equations are considered to be used for the first level of mapping between a SWEBOK and CBOK (i.e. between knowledge areas).

Step 1: Application of union to SWEBOK and CBOK knowledge areas:

The *union* of *S* and *C*, written $S \cup C$, is the set whose elements are just the elements of *S* or *C* or of both. The union operation uses the following equation to illustrate the knowledge areas that are described in the SWEBOK or in the CBOK.

This operation is defined in the predicate notation as follow:

$$S \cup C == \{e | e \in S \vee e \in C\}$$

$$S \cup C ==$$

{SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF, MF,EF}

$\cup$

{EPC,SYE,RE,SD,SC,T,SM,CM,SEM,SEP,SQ,MF,CF,SE}

The union relation between *set S* and *set C* the following new set is derived and these include all the knowledge areas that are incorporated in both *set S* and *set C*. The result is described in *set* $S \cup C ==$ the union result between SWEBOK (S) and CBOK (C).

In this union result the naming of the SWEBOK Guide knowledge areas is adopted. For instance, software requirement in SWEBOK instead of requirements engineering in CBOK.

$S \cup C ==$*{SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SYE,EF,SEE,CF,MF,SE}*

Table 1 describes from left to right the SWEBOK knowledge areas, the CBOK knowledge areas and the related union that shows the results of the knowledge areas that are incorporated in both the SWEBOK or in the CBOK.

Table 1: SWEBOK and CBOK mapping union and intersection results

SWEBOK	CBOK	SWEBOK $\cup$ CBOK	SWEBOK $\cap$ CBOK
Software requirements	Requirements engineering	Software requirements or Requirements engineering	Software requirements or Requirements engineering
Software design	Software design	Software design	Software design
Software construction	Software construction	Software construction	Software construction
Software testing	Testing	Softwaretesting or Testing	Software testing or Testing
Software maintenance	Software maintenance	Software maintenance	Software maintenance
Software configuration management	Configuration management	Software configuration management or configuration management	Software configuration management
Software engineering management	Software engineering management	Software engineering management	Software engineering management
Software engineering process	Software engineering process	Software engineering process	Software engineering process

Table 1: SWEBOK and CBOK mapping union and intersection results (Contd.)

SWEBOK	CBOK	SWEBOK $\cup$ CBOK	SWEBOK $\cap$ CBOK
Software engineering methods		Software engineering methods	
Software quality	Software quality	Software quality	Software quality
Software engineering professional practice	Ethics and professional conduct	Software engineering professional practice or Ethics and professional conduct	Software engineering professional practice or Ethics and professional conduct
Engineering economy foundations		Engineering economy foundations	
Computing foundations	Computing fundamentals	Computing foundations	Computing fundamentals
Mathematical foundations	Mathematics fundamentals	Mathematical foundations	Mathematics fundamentals
Engineering foundations		Engineering foundations	
	System engineering	System engineering	
	Software engineering	Software engineering	

Step 2: Application of intersection to the SWEBOK and the CBOK knowledge areas

The *intersection* of *S* and *C*, written $S \cap C$, is the set whose elements are just the elements of both *S* and *C*. The intersection operation uses the subsequent equation to illustrate the knowledge areas that are described both in the SWEBOK Guide and in the CBOK.

This operation is defined in the predicate notation as follow:

$$S \cap C == \{e | e \in S \wedge e \in C\}$$

$$S \cap C ==$$

{SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF, MF,EF}

$\cap$

{EPC, SYE, RE, SD, SC, T, SM, CM, SEM, SEP, SQ, MF, CF, SE}

The intersection relation between *S* and *C* the result includes the knowledge areas that are present in both *S* and *C*. The result is described in *set* $S \cap C ==$ the intersection result between SWEBOK (S) and CBOK (C).

$S \cap C ==$*{SR, SD, SC, ST, SM, SCM, SEM, SEP, SQ, SEPP}*

Table 1 describes from left to right the SWEBOK knowledge areas, the CBOK knowledge areas and the related intersection

that shows the knowledge areas that are present in both the SWEBOK and in the same time in the CBOK.

Step 3: Application of the difference to the SWEBOK and the CBOK knowledge areas:

The difference between the SWEBOK and CBOK guides means the *relative complement* of *C* in *S* (also called the *set-theoretic difference* of *S* and *C*), denoted by *S* \ *C*, (or *S* − *C*) is the set of all elements which are members of *S* but not members of *C*; (i.e. {1, 2, 3, 4} \ {1, 3} = {2, 4}).

The deference operation uses the equation below to show the knowledge areas that are described in the SWEBOK and are not described in the CBOK.

These operations define the predicate notation as follows:

$$C - S == \{e | e \in C \wedge e \notin S$$

$$S - C == \{e | e \in S \wedge e \notin C\}$$

S - C == *{SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF, MF,EF}*

—

{EPC, SYE, RE, SD, SC, T, SM, CM, SEM, SEP, SQ, MF, CF, SE}

The difference relation between *S* and *C* the following new set includes all the knowledge areas, which are described in *S* are not described in *C*. The result is described in set $S - C$ the knowledge areas that are described in S and not in C.

$S - C$== *{SEME, EEF, EF}*

The difference relation between C and S the following new set includes all the knowledge areas, which are described in S are not described in C. The result is described in set $C - S$: the knowledge areas that are described in C and not in S.

$C - S$:== *{SYE, SE}*

Table 2 describes from left to right the SWEBOK knowledge areas, the CBOK knowledge areas, the related differences that shows the knowledge areas that are present in the SWEBOK and are not present in the CBOK and the knowledge areas that are present in the CBOK and not in the SWEBOK.

Table 2: SWEBOK and CBOK difference results

SWEBOK	CBOK	SWEBOK – CBOK (S-C)	SWEBOK – CBOK (C-S)
Software requirements	Requirements engineering		
Software design	Software design		
Software construction	Software construction		
Software testing	Testing		
Software maintenance	Software maintenance		
Software configuration management	Configuration management		
Software engineering management	Software engineering management		

Table 2: SWEBOK and CBOK difference results **(Contd.)**

SWEBOK	CBOK	SWEBOK – CBOK (S-C)	SWEBOK – CBOK (C-S)
Software engineering process	Software engineering process		
Software engineering methods		Software engineering methods	
Software quality	Software quality		
Software engineering professional practice	Ethics and professional conduct		
Engineering economy foundations		Engineering economy foundations	
Computing foundations	Computing fundamentals		
Mathematical foundations	Mathematics fundamentals		
Engineering foundations		Engineering foundations	
	System engineering		System engineering
	Software engineering		Software engineering

IV. Mapping between SWEBOK Guide and CBOK using at Subarea Level

A. Phase 1: Initiation mapping phase

The following section present the formal mapping between the SWEBOK and the CBOK bodies of knowledge at the sub area level. For the purpose of this study just software requirements will be taken as example for the initiation and practical phase description; however the complete analysis at subarea level will be covered in result phase. In the initiation phase the two subsets will be defined for instance, the list of software requirement subareas for the SWEBOK and requirements engineering of the CBOK with their related symbols. In the practical phase the three basic operations that belong to set theory such as: union, intersections and difference will be applied to the list of software requirements subareas.

Step 1: Subsets definition for SWEBOK and CBOK at subareas level**:**

In this section, the SWEBOK software requirements knowledge area and the CBOK requirements engineering knowledge area are to be considered as object of this formal mapping. In the SWEBOK the software requirements subareas subset are defined as follows:

Software Requirements (SWEBOK) = {Software requirements fundamentals (SRF), requirements process (RP), requirements elicitation (RE), requirements analysis (RA), requirements specification (RS), requirements

validation (RV), practical considerations (PC), Software requirement tools (SRT)}

The subset of subareas in requirements engineering in the CBOK Guide consists of the following:

Requirements engineering (CBOK) = {Fundamentals of requirements engineering (FRE), requirements engineering process (REP), initiation and scope definition (ISD), requirements elicitation (RE), requirements analysis (RA), requirements specification (RS), requirements validation (RV), practical considerations (PC)}

Step 2: Symbolize subsets definition for SWEBOK and CBOK subarea level:

This part illustrates both the software requirements of the SWEBOK and requirements engineering of the CBOK sets using symbols. The following is SR_s containing subareas from the software requirements of the SWEBOK and RE_c requirements engineering of the CBOK.

SR_s == *{SRF, RP, RE, RA, RS, RV, PC, SRT}*

RE_c == *{FRE, REP, ISD, RE, RA, RS, RV, PC}*

Step 3: Sets associations at second level of mapping:

Figure 6 illustrates the visual associations that exist between the SWEBOK software requirements and the requirements engineering CBOK at subareas level. The interconnected subareas demonstrate the regular subareas that are present in both software requirements and in requirements engineering. For instance, software requirements fundamentals (SRF) in SWEBOK are the same as in the CBOK but with a different name, fundamentals of requirements engineering (FRE).

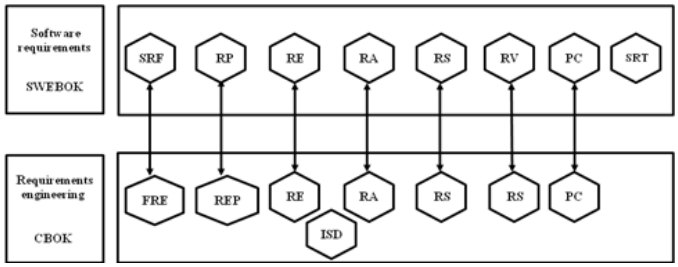

Fig. 6: Visual association between software requirements second level of breakdown

B. Phase 2: Application mapping phase at subarea level

An example of the formal mapping that is based on a set theory is illustrated at the second level of subareas for both the SWEBOK and the CBOK guides (i.e. between subareas).

Step 1: Application of union mapping at sub Area level for software requirement or requirements engineering knowledge area**:**

The *union* of SR_s and RE_c, written $SR_s \cup RE_c$, is the set whose elements are the elements of SR_s or RE_c or of both. The union operation uses the equation described below; this will be used to categorize the subareas that are described in the software requirements in the SWEBOK or in the requirements engineering of the CBOK.

In the predicate notation the definition is:

$$SRs \cup REc = \{e | e \in SRs \vee e \in REc\}$$

SR_s == *{SRF, RP, RE, RA, RS, RV, PC, SRT}*

$\cup$

RE_c == *{FRE, REP, ISD, RE, RA, RS, RV, PC}*

The following result is based on the union relation between SR_s and RE_c and it's includes all the subareas that are presents in SR_s or RE_c. $SR_s \cup RE_c$ defines the result of the union operation between SR_s or RE_c. In this union result the description of SR_s element are adopted.

$SR_s \cup RE_c$ == *{SRF, RP, ISD, RE, RA, RS, RV, PC, SRT}*

Appendix 1 illustrates the result of the formal mapping using union operation between the related subareas of the SWEBOK and the related subareas of the CBOK.

Step 2: Application of intersection mapping at subarea level for software requirement or requirements engineering knowledge area

The *intersection* of SR_s and RE_c, written $SR_s \cap RE_c$, is the set whose elements are just the elements of both SR_s and RE_c. The intersection operation uses the equation described below; this will be used to group the subareas that are defined in both the software requirements in the SWEBOK and in the requirements engineering of the CBOK.

This operation defined the predicate notation as follow:

$$SRs \cap REc = \{e | e \in SRs \wedge e \in REc$$

SR_s== *{SRF, RP, RE, RA, RS, RV, PC, SRT}*

$\cap$

RE_c== *{FRE, REP, ISD, RE, RA, RS, RV, PC}*

The following result is based on the intersection relation between SR_s and RE_c and it's includes all the subareas that are included in both SR_s and RE_c. $SR_s \cap RE_c$ defines the result of the intersection operation between SR_s and RE_c. In this intersection result the descriptions of the SR_s subareas are adopted.

$SR_s \cap RE_c$ == *{SRF, RP, RE, RA, RS, RV, PC}*

Appendix 1 illustrates the result of the formal mapping using intersection operation between the related subareas of the SWEBOK and the related subareas of the CBOK.

Step 3: Application of difference operation at subarea level for software requirements or requirements engineering knowledge area:

The difference between SR_s and RE_c means the *relative complement* of *REc* in SR_s (also called the *set-theoretic difference* of SR_s and RE_c, denoted by $SR_s \setminus RE_c$, or $SR_s - RE_c$) is the set of all elements which are members of SR_s but not members of RE_c.

These operations define the predicate notation as follows:

$$SRs - REc = \{e | e \in SRs \wedge e \notin REc\}$$

$$REc - SRs = \{e | e \in REc \wedge e SRs\}$$

$SR_s - RE_c$ == *{SRF, RP, RE, RA, RS, RV, PC, SRT}*

—

{FRE, REP, ISD, RE, RA, RS, RV, PC}

The difference relation between SR_s and RE_c the resulting subset includes all the subareas, which are defined in SR_s and are not defined in RE_c. The result is described in:

$SR_s - RE_c$

$SR_s - RE_c$ == *{SRT}*

The difference relation between subsets RE_c and SR_s the resulting subset includes all the subareas, which are defined in RE_c and are not defined in SR_c. The result is described in $RE_c - SR_s$.

$RE_c - SR_s$ == *{ISD}*

Appendix 2 describes the results of the difference operation between SWEBOK subareas and CBOK subareas from left to right.

Figure 7 summarizes the formal mapping approach for integrated CBOK with software engineering body of knowledge SWEBOK and their subareas as follows:

The Generalization set of knowledge areas in Software Engineering for
SWEBOK ∧ / ∨ CBOK = {e|e ∈ S ∨ e ∈ C} ... (1)
The Generalization set of Sub-knowledge areas in Software Engineering for
SR_s == {SRF, RP, RE, RA, RS, RV, PC, SRT} ... (2)
RE_s == {FRE, REP, ISD, RE, RA, RS, RV, PC} ... (3)
The Generalization set of the UNIOUN between the knowledge areas in Software Engineering for SWEBK and CBOK:
S ∪ C == {e|e ∈ S ∨ e ∈ C} ... (4)
The Generalization set of the UNIOUN between Sub-knowledge areas in Software Engineering for SWEBK and CBOK:
SRs ∪ REc = {e|e ∈ SRs ∨e ∈ REc} ... (5)
The Generalization set of the INTERSECTION between the knowledge areas in Software Engineering for SWEBK and CBOK:
S ∩ C == {e|e ∈ S ∧ e ∈ C} ... (6)
The Generalization set of the INTERSECTION between Sub-knowledge areas in Software Engineering for SWEBK and CBOK:
SRs ∩ REc = {e|e ∈ SRs ∧ e ∈ REc} ... (7)
The Generalization set of the DIFFERENCE between the knowledge areas in Software Engineering for SWEBK and CBOK:
S − C == {e|e ∈ S ∧ e ∉ C} ... (8)
C − S == {e|e ∈ C ∧ e ∉ S} ... (9)
The Generalization set of the DIFFERENCE between the Sub-knowledge areas in Software Engineering for SWEBK and CBOK:
SRs − REc = {e|e ∈ SRs ∧ e ∉ REc} ... (10)
REc − SRs = {e|e ∈ REc ∧ e SRs} ... (11)

Fig. 7 Formal mapping approach for integrated CBOK with SWEBOK

V. Results and Analysis Phase

A. First Level of mapping: Knowledge area

The formal mapping of the SWEBOK and the CBOK using basic operations set theory at knowledge area level showed the following results in terms of breadth as is described in figure 8.

- Over the total of 17 knowledge areas that belong to the SWEBOK or to the CBOK using the union operation. Among them 15 KA's are present in the SWEBOK and 14 KA's are present in the CBOK.
- The total of 12 KA's that belong to both the SWEBOK and the CBOK using the intersection operation.
- The total of three KA's that belong to the SWEBOK and don't belong to the CBOK using the difference operation and also two KA's that belong to the CBOK and don't belong to the SWEBOK.

This result covered all possibilities for the mapping and this by using the basic set theory operations.

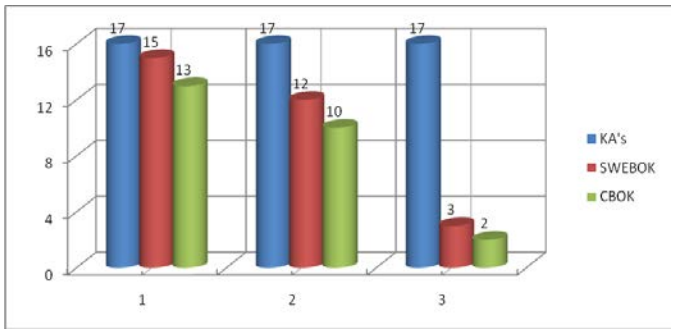

Fig. 8: Mapping result at knowledge area

This formal mapping showed the following differences concerning the KAs characterizing the practice of software engineering in the SWEBOK Guide and the core knowledge areas of the CBOK are listed as follow:

- The identification of the following knowledge areas software requirements, software testing, software configuration management and software engineering professional practice as adopted in the SWEBOK is modified in the same order to requirements engineering, testing, configuration management and ethics and professional conducts.
- System engineering knowledge area is described in the CBOK but not in the SWEBOK.
- The software engineering models and methods KA is included in the SWEBOK Guide, but not in the CBOK.

The differences associated to the educational requirements of software engineering in the SWEBOK Guide and the preparatory knowledge areas in the CBOK are described as follow:

- Engineering foundations is included in the SWEBOK Guide, but not in the CBOK.
- Computing fundamentals and mathematical fundamentals in the CBOK are considered as preparatory knowledge areas. In the SWEBOK Guide, they are considered as educational requirements of software engineering, with slightly different names: computing foundations and mathematical foundations.
- The software engineering knowledge area is included as a preparatory knowledge area of the CBOK. It defines

the basic knowledge that students should possess in software engineering when entering a Master‘s degree program. This knowledge area is not included in the SWEBOK Guide.

The differences related to both the core knowledge areas in the CBOK and the educational requirements of software engineering in the SWEBOK Guide are the following:

- Engineering economics is considered an educational requirements KA of software engineering in the SWEBOK Guide but is not covered in the CBOK at least not at the knowledge area level.

The SWEBOK and the CBOK share identical broad knowledge areas that characterizes the practice of software engineering in the SWEBOK and a core body of knowledge for the CBOK: software requirements or requirements engineering, software design, software construction, software testing or testing, software maintenance, software configuration management or configuration management, software engineering management, software engineering process, software quality and software engineering professional practice or ethics and professional conduct.

An additional two Knowledge areas characterize the educational requirements of software engineering for the SWEBOK and a preparatory knowledge for the CBOK: computing foundations or computing fundamentals, mathematical foundations or mathematical fundamentals.

In terms of breadth the scope of the new version of the SWEBOK is wider than the scope of the CBOK. the result showed that the SWEBOK is the primarily source for developing the CBOK covering 10 out of 11 KA’s from the practice of software engineering in the SWEBOK and two out of four KA's from the educational requirements KA's of the SWEBOK.

B. Second Level of mapping: SubAreas Level

The results of the mapping depicted in appendix 1 between the SWEBOK and CBOK at the second level of breakdown is divided into three categories as follows: the first category illustrates name changes in the related subareas, the second category describes the difference between SWEBOK and CBOK represented by (S-C) that includes the subareas that are present in SWEBOK and not in CBOK. The third category highlights the difference between the CBOK and the SWEBOK represented by (C-S) that includes the subareas that are present in CBOK and not in SWEBOK.

This mapping showed the following differences regarding the KAs characterizing the practice of software engineering in the SWEBOK Guide and the core knowledge areas of the CBOK:

- Six of the subareas covered have different names. Two of them have a different name plus additional information such as: The SWEBOK doesn't cover project organization in software project enactment subarea and the CBOK doesn't cover improvement in software engineering process in process assessment subarea.
- Eight subareas are missing in the SWEBOK guide.
- Eighteen subareas are missing in the CBOK guide.

Category 1: Changes in naming

This section highlights the different name changes made from the SWEBOK to the CBOK in subareas description without looking into the content, these results are presented by * in table 1: Software requirement:

- Software requirements fundamentals and requirements process subareas in the software requirements of the SWEBOK are changed to fundamentals of requirements engineering and to requirements engineering process in requirements engineering of the CBOK.
- Software testing fundamentals subarea in the SWEBOK is changed to testing fundamentals in the CBOK.
- Management of the SCM process, software configuration identification, software configuration control and software configuration status accounting are changed in the CBOK as follows to: management of the CM process, configuration identification, configuration control and configuration status accounting. The word software is removed from the titles of the subareas.
- Software project enactment subarea in software engineering management of the SWEBOK is changed to software project organization and enactment in software engineering management of the CBOK.
- Software measurement subarea in software engineering process in the SWEBOK is changed in the CBOK to product and process measurement subarea.
- Software process definition subarea in software engineering process in the SWEBOK is changed to process definition in the CBOK.
- Software process assessment and improvement in software engineering process in the SWEBOK is changed to process assessment.
- Professionalism in software engineering professional practice in the SEWBOK is named as codes of ethics and professional conduct in the CBOK and it is included as a topic as well in the SWEBOK.

Category 2: SWEBOK- omitted subareas

This section illustrates the missing subareas in the SWEBOK that represent the difference of subareas between CBOK and SWEBOK (C-S). That is the subareas that are added to the CBOK:

- Iinitiation and scope definition subarea in requirements engineering KA.
- Risk management and engineering economics subareas in software engineering management KA.
- Process implementation and change in software engineering process
- Verification and validation subarea in software quality KA.
- Social, legal, and historical issues, code of ethics and professional conduct and the nature role of software engineering standards subareas in ethics and professional conduct KA.

Category 3: CBOK- omitted subareas

Several subareas were added in the actual version of the SWEBOK. The content of the CBOK doesn’t include some of these subareas. This section presents the different subareas

that are added to the SWEBOK and missing in the CBOK, this includes the following:

- Software requirements tools in software requirements KA.
- User interface design, software design tools subarea in software design.
- Construction technologies and construction tools in software construction KA.
- Software testing tools in software testing KA.
- Software maintenance tools in software maintenance KA.
- Software configuration auditing and software configuration management tools in Software configuration management KA.
- Initiation and scope definition and software engineering management tools in software engineering management KA.
- Software life cycles and software engineering process tools in the software engineering process KA.
- Software quality practical considerations and software quality tools in software quality KA.
- Professionalism, group dynamics and psychology, communications skills in software engineering professional practices KA.

This mapping showed the related differences of the educational requirements of software engineering in the SWEBOK Guide and the preparatory knowledge areas in the CBOK:

Category 1: CBOK- omitted subareas

- The following set of subareas is added to the computing foundations of the SWEBOK: problem solving techniques, abstraction, programming language basics, debugging tools and techniques, basic concept of a system, compiler basics, database basics and data management, parallel and distributed computing, basic user human factors, basic developers human factors, secure software development and maintenance.
- Sets relations, functions, basic logic, proof techniques, basic counting, graph and trees, discrete probability are added to mathematical foundations of the SWEBOK as subareas.
- Finite state machine, grammars, numerical precision, accuracy and errors, number theory and algebraic structure are added to mathematical foundations of the SWEBOK.

Category 2: SWEBOK- omitted subareas

- Module design and construction is added to the computing fundamentals of the CBOK.
- Discrete structure, propositional and predicate logic, probability and statistics are added to the mathematics fundamentals of the CBOK.

Category 3: SWEBOK vs. CBOK name substitution

- Computer architecture subarea in the CBOK is described as computer organization in the SWEBOK.
- Data structures and algorithms subarea in computing fundamentals in the CBOK is described in two different subareas such as data structures and representation, algorithm and complexity in the SWEBOK.
- Operating system in computing fundamentals in the CBOK is described as operating system basics in computing foundations in the SWEBOK.
- Network and communication in computing fundamentals is described in the CBOK as network and communication basics in computing foundations in the SWEBOK.

C. Third Level: SWEBOK vs. CBOK Topics Mapping Results

This section describes the mapping at the third level of breakdown between the SWEBOK version 3 and the CBOK version 1. The result of the differences is exposed by each knowledge area characterizing the practice of software engineering in the SWEBOK Guide and the core knowledge areas of the CBOK:

1) Software requirements or requirements engineering knowledge area (KA)

- Definition of a software requirement under software requirements fundamentals in the SWEBOK is named definition of requirements in CBOK.
- Addition of system requirements and software requirements under software requirements fundamentals subarea in SWEBOK.
- Addition of architectural design and requirements allocation and formal analysis under requirements analysis subarea in the SWEBOK.
- Addition of system definition document, systems requirement specification and software requirement specification topics under requirements specification subarea in the SWEBOK.
- Addition of requirements specification techniques under requirement specification subarea in the CBOK.
- Addition of relationship between systems engineering and software engineering, system design constraints and system design and requirements allocation topics under fundamentals of requirements engineering subarea in CBOK.
- Addition of initiation and scope definition subarea in CBOK.
- Addition of heuristics methods and formal methods topics under requirement analysis in CBOK.

2) Software design

- Addition of hardware issues topic under software structure and architecture subarea in the SWEBOK.
- Addition of data-structured centered design and other methods topics under software designs strategies and methods subarea in SWEBOK.
- Addition of heuristics methods and formal methods topics under software designs strategies and methods subarea in CBOK.

3) Software Construction

- Addition of executable models topic under practical considerations subarea in the SWEBOK.

4) Software Testing

- Addition of software testing tools topic under test process in the SWEBOK.
- Addition of practical considerations topic in the SWEBOK under test process.
- Addition of system testing and software testing under testing fundamentals subarea in the CBOK.
- Addition of component testing, integration testing, system testing and acceptance testing topics under test levels subarea in the CBOK.
- Addition of management concerns topic under test process subarea in the CBOK.

5) Software Maintenance

- Maintenance processes topic under maintenance process subarea in SWEBOK is replaced by maintenance process models topic under maintenance process in CBOK.
- Migration and retirement topics under techniques for maintenance in SWEBOK are not present in CBOK under techniques for maintenance.
- Technical issues topic under key issues in software maintenance in the SWEBOK is renamed technical in CBOK under key issues in software maintenance.

6) Software configuration management

- SCM plan under management of the SCM process in the SWEBOK is changed to configuration management plan under management of the CM process in the CBOK.
- Requesting, evaluating and approving software changes under software configuration identification in the SWEBOK is changed to requesting, evaluating and approving changes under configuration control in the CBOK.
- Implementing software changes under software configuration identification in the SWEBOK is changed to implementing changes under configuration control in the CBOK.
- Software configuration status reporting under software configuration status accounting 4.in the SWEBOK is replaced with configuration status reporting under software configuration status accounting 4 in the CBOK.
- Software configuration status information topic under software configuration status accounting is not present in the CBOK under configuration status accounting.

7) Software engineering management

- Determine deliverables under software project planning in the SWEBOK is replaced with projects deliverables under software project planning in the CBOK.
- Risk management and plan management under software project planning in the SWEBOK are omitted in the CBOK under software project planning.
- Project goals and objectives, Project policies and standards, Project assumptions and forecasts, project staffing, project plan/budget development and management topics under software project planning in the CBOK doesn't exist in the SWEBOK under software project planning.
- Project organization, project directing, project control and supplier contract management (e.g., RFP, cost evaluation, IP rights) topics in the CBOK under software project organization and enactment in the CBOK doesn't exist in the SWEBOK under software project enactment.
- Implementation of plans, software acquisition and supplier contract management, implementation of measurement process, control process, monitor process and reporting topics under software project enactment in the SWEBOK are not present in the CBOK under software project organization and enactment subarea.

8) Software engineering Process

- Software process management and software process infrastructure topics under software process definition in the SWEBOK are not present in the CBOK under process definition.
- Software process improvement models and continuous and staged ratings topics under software process assessment and improvement in the SWEBOK are omitted in the CBOK under process assessment.
- Software process and product measurement topic under software measurement in the SWEBOK is substituted to two different topics as follows: software process measurement and software product measurement in the CBOK under product and process measurement.
- Software information models topic under software measurement subarea in the SWEBOK is omitted in the CBOK under product and process measurement.
- Software process measurement techniques topic under software measurement in the SWEBOK is changed to measurement techniques in CBOK under product and process measurement.
- Life cycles models, software life cycle processes notations for process definitions, process adaptation and automation topics under process definition in the CBOK are omitted in the SWEBOK under software process definition.

9) Software quality

- Models and quality characteristics, software quality improvements topics under software quality fundamentals in the SWEBOK are substituted to quality models and characteristics, quality

improvements topics under software quality fundamentals in the CBOK.

- Software safety topic under software quality fundamentals in the SWEBOK is omitted in the CBOK under software quality fundamentals.
- Verification and validation, reviews and audit in the SWEBOK under software quality, management processes are omitted in the CBOK under software quality management processes.
- Application quality requirements and defect characterization topics in the CBOK under software quality fundamental are omitted in the SWEBOK under software quality fundamentals.
- Software quality management techniques and software quality measurement topics under software quality management processes in the CBOK are omitted in the SWEBOK under software quality management processes.

10) Software engineering professional practices

- All the subareas defined in the SWEBOK are completely different than the ones defined in the CBOK.

The differences related to the topics of knowledge areas characterizing the educational requirements of software engineering in the SWEBOK and the preparation knowledge for the core body of knowledge.

11) Computing foundations of the SWEBOK and computing fundamentals of the CBOK

- Overview of programming languages; virtual machines; introduction to language translation; declaration and types; abstraction mechanisms; object-oriented programming; functional programming; language translation systems; type systems; programming language semantics; programming language design topics under programming fundamentals of the CBOK are omitted in the SWEBOK under programming fundamentals.
- Concurrency, scheduling and dispatch, memory management, device management, security and protection, file systems, real-time and embedded systems, fault tolerance,
- System performance evaluation, scripting topics under operating systems in the CBOK are omitted in the SWEBOK under operating systems basics.
- Operating system overview and principles under operating systems in the CBOK is replaced by operating systems overview in the SWEBOK under operating systems basics.
- Introduction to net-centric computing, communication and networking, network security, building Web applications, network management, compression and decompression, multimedia data technologies, wireless and mobile computing topics under networks and communications in the CBOK are omitted in the SWEBOK under networks and communications basics.
- Machine level representation of data, assembly level, machine organization, interfacing and communication, functional organization, multiprocessing and alternative architectures, performance enhancements, architecture for networks and distributed systems topics under computer architecture in the CBOK are omitted in the SWEBOK under computer organization.
- Digital logic and digital systems, topic in the CBOK under computer architecture is divided into two separated topics such as digital systems and digital logic in the SWEBOK under computer organization.
- Memory system organization and architecture topic under computer architecture in the CBOK is replaced with memory system organization in the SWEBOK under computer organization.
- The programming process, programming paradigm and defensive programming topics under programming fundamentals in the SWEBOK are omitted in the CBOK under programming fundamentals.
- Computer organization overview, computer expression of data, the central processing unit (CPU), input and output (I/O) topics in the SWEBOK under computer organization are omitted in the CBOK under computer architecture.
- Tasks of operating systems, operating system abstraction and operating system classification topics in the SWEBOK under operating systems basics are omitted in the CBOK under operating systems.
- Types of networks, basic network components, networking protocols and standards, internet of things and virtual private network topics in the SWEBOK under network communication basics are omitted in the CBOK under networks and communications.
- Data structure overview, types of data structure and operation on data structures topics under data structure and representation in the SWEBOK are omitted in the CBOK under data structures and algorithms.
- Overview of algorithms attributes of algorithms topics in the SWEBOK under algorithms and complexity are omitted in the CBOK under data structures and algorithms.
- Algorithmic analysis topic in the SWEBOK under algorithms and complexity is renamed in the CBOK basic algorithms analysis under data structures and algorithms.
- Algorithmic analysis strategies and algorithmic design strategies topics in the SWEBOK under algorithm and complexity is replaced by algorithmic strategies in the CBOK under data structures and algorithms.

Fundamentals of computing algorithms and distributed algorithms topics in the CBOK under data structures and algorithms are omitted in the SWEBOK under algorithms and complexity.

Figure 8 summarizes the formal mapping approach results for integrated CBOK with software engineering body of knowledge SWEBOK and their subareas

Results of ... **(1)**

SWEBOK (S)= {Software requirements (SR), Software design (SD), Software construction (SC), Software testing (ST), Software maintenance (SM), Software configuration management (SCM), Software engineering management (SEM), Software engineering process (SEP), Software engineering methods (SEMM), Software quality (SQ), Software engineering professional practice (SEPP), Software Engineering economics (SEE), Computing foundations (CM), Mathematical foundations (MF) and Engineering foundations (EF)}

CBOK (C) = {Ethics and professional conduct (EPC), System engineering (SYE), Requirements engineering (RE), Software design (SD), Software construction (SC), Testing (T), Software maintenance (SM), Configuration management (CM), Software engineering management (SEM), Software engineering process (SEP), Software quality (SQ), Mathematical fundamentals, Computing fundamentals, software engineering (SE)}

Formal Results of... **(1)**

The following is set S containing knowledge areas from the SWEBOK

S= {SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF,MF,EF}

The following is set C containing knowledge areas from the CBOK.

C= {EPC, SE, RE, SD, SC.T, SM, CM, SEM, SEP, SQ}

Formal Results of... **(4)**

$S \cup C$ = {SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF,MF,EF}∪ {EPC,SYE,RE,SD,SC,T,SM,CM,SEM,SEP,SQ,MF,CF,SE}

$S \cup C$ = {SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SYE,EF,SEE,CF,MF,SE}

Formal Results of... **(5)**

SR_s − {SRF, RP, RE, RA, RS, RV, PC, SRT} ∪ RE_c − {FRE, REP, ISD, RE, RA, RS, RV, PC}

$SR_s \cup RE_c$ = {SRF, RP, ISD, RE, RA, RS, RV, PC, SRT}

Formal Results of... **(6)**

$S \cap C$ = {SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF,MF,EF} ∩ {EPC, SYE, RE, SD, SC, T, SM, CM, SEM, SEP, SQ, MF, CF, SE}

$S \cap C$ = {SR, SD, SC, ST, SM, SCM, SEM, SEP, SQ, SEPP}

Formal Results of... **(7)**

SR_s= {SRF, RP, RE, RA, RS, RV, PC, SRT} ∩ RE_c= {FRE, REP, ISD, RE, RA, RS, RV, PC}

$SR_s \cap RE_c$ = {SRF, RP, RE, RA, RS, RV, PC}

Formal Results of... **(8)** *and (9)*

$S - C$ = {SR,SD,SC,ST,SM,SCM,SEM,SEP,SEMM,SQ,SEPP,SEE,CF,MF,EF} - {EPC, SYE, RE, SD, SC, T, SM, CM, SEM, SEP, SQ, MF, CF, SE}

The result is described in set $S - C$.

$S - C$= {SEME, EEF, EF}

The result is described in set $C - S$:

$C - S$:= {SYE, SE}

Formal Results of... **(10)** *and (11)*

The result is described in set SRs - REc

$SR_s - RE_c$ = {SRF, RP, RE, RA, RS, RV, PC, SRT} − {FRE, REP, ISD, RE, RA, RS, RV, PC}

$SR_s - RE_c$ − {SRT}

$RE_c - SR_s$ = {ISD}

Fig. 8 the formal mapping approach results for integrated CBOK with SWEBOK

VI. Conclusion and Future Work

Software engineering discipline lacks formal mapping methods; in the up to date literature all the mapping are done in a non formal way therefore giving imprecise results and prone to mistakes.

The SWEBOK is considered as a support for software engineering program curriculums and certifications. The GSwE2009 curriculum includes the core body of knowledge CBOK which provides guidelines for the education of students for a professional master's degree in software engineering.

This study proposes a three phase's formal mapping methodology based on set theory operations: initiation, practical, results and analysis phases. This paper takes as object of study two bodies of knowledge SWEBOK and CBOK guide and explores them in terms of breadth and in terms of depth through the decomposition break down topics; where similarities and differences are highlighted.

This work has showed that set theory was successfully used to map the software engineering body of knowledge (SWEBOK) and core body of knowledge (CBOK). Therefore it can be used in software engineering discipline for the purpose of a formal mapping. This mapping provided the differences and similarities using a formal approach that give a rigorous result and allow the reduction of errors that may be very common using a conventional mapping. This result also shows that there is no overwhelming differences school of thought between the two guides. Future work will concentrate on the implementation of an automated tool based on formal mapping approach proposed in this study.

References

[1] Kenza Meridji, Khaled Al Sarayreh "Mapping the SWEBOK Guide to the Core Body of Knowledge of GSWE2009 Using Set Theory" WSEAS

[2] Kenza Meridji, Alain Abran "Software Engineering Principles: Do they meet Engineering Criteria?", Journal of Software Engineering & Applications, Vol.3 No. 10 , October 2010

[3] Wang Y. (2007). Software Engineering Foundations: A Software Science Perspective. New York Auerbach publications

[4] IEEE Std 610.12, IEEE Standard Glossary of Software Engineering Terminology. Corrected Edition, February, 1991.

[5] P. Bourque, Richard E. Fairley "Guide to the Software Engineering Body of Knowledge (SWEBOK)": Version 3.0, IEEE Computer Society Press Los Alamitos, CA, USA

[6] A. Abran, J. W. Moore (exec. eds), P. Bourque and R. Dupuis (eds.), Guide to the Software Engineering Body of Knowledge (SWEBOK). IEEE Computer Society Press, 2004.

[7] Normand Seguin, Alain Abran, Robert Dupuis "Software Engineering Principles: A Survey and an Analysis" C3S2E '10 Proceedings of the Third Conference on Computer Science and Software Engineering, Pages 59-65. ACM New York

[8] Integrated Software & Systems Engineering Curriculum (iSSEc) Project, Graduate Software Engineering 2009 - Curriculum Guidelines for Graduate Degree Programs in Software Engineering (GSwE2009), www.gswe2009.org.

[9] ISO-TR-19759 (2005) Software Engineering — Guide to the Software Engineering Body of Knowledge (SWEBOK), International Organization for Standardization, Switzerland.

[10] M. Ardis and G. Ford, SEI Report on Graduate Software Engineering Education, CMU/SEI 89-TR-21, Software Engineering Institute, Carnegie Mellon University, June 1989.

[11] ACM/IEEE-CS Joint Task Force on Computing Curricula, Software Engineering 2004: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, August 2004, http://www.acm.org/education/curricula-recommendations.

[12] P. Bourque, F. Robert, J.-M. Lavoie, A. Lee, S. Trudel, T. Lethbridge, Guide to the Software Engineering Body of Knowledge (SWEBOK) and the Software Engineering Education Body of Knowledge (SEEK) – A Preliminary Mapping. Proceedings of the 10th International Workshop on Software Technology and Engineering Practice (STEP02) IEEE Computer Society.

[13] M. Barker and K. Inoue (2009). IT SPIRAL: A Case Study in Scalable Software Engineering Education. Software Engineering Education and

Training, 2009. CSEET '09. 22nd Conference on Software Engineering Education and Training, 2009.

[14] A. Comer and H. M. Edwards (2006). Students Managing the Software Development Process: A Meta-Level Retrospective Evaluation. Software Engineering Education and Training, 2006. Proceedings. 19th Conference on. Software Engineering Education & Training (CSEET'06), 2006.

[15] J. M. Fernandes and R. J. Machado (2006). A Two-Year Software Engineering M.Sc. Degree Designed Under the Bologna Declaration Principles. Software Engineering Advances, International Conference on. Software Engineering Advances (ICSEA'06), Papeete, Tahiti, French Polynesia, 2006 – "Proceedings of the International Conference on Software Engineering Advances." [S.l.]: IEEE Computer Science Press, 2006. ISBN 0-7695-2703-5.

[16] A. Pyster, R. Turner, D. Henry, K. Lasfer, L. Bernstein., K. Baldwin (2008). The Current State of Software Engineering, Master's Degree Programs. Software Engineering Education and Training, 2008. CSEET '08. IEEE 21st Conference on. Software Engineering Education and Training.

[17] H. L. Welch, (2007). Teaching a service course in software engineering. Frontiers In Education Conference -- Global Engineering: Knowledge Without Borders, Opportunities Without Passports, 2007. FIE '07. 37th Annual.

First A. Kenza Meridji is an Assistant Professor at Petra University Amman, Jordan. She is pursuing research on the fundamental principles of software engineering and non-functional requirements. She holds a PhD degree from the École de Technologie Supérieure – Université du Québec, Montreal, Canada and a Master degree in Software Engineering from Concordia University, Canada.

Appendix 1: SWEBOK Guide and CBOK union, and intersection results at subarea level

SWEBOK SUBAREAS	CBOK SUBAREAS	SWEBOK ∪ CBOK SUBAREAS	CBOK ∩ SWEBOK SUBAREAS
Software Requirements	Requirements Engineering	-Software Requirements Fundamentals Or Fundamentals Of Requirements Engineering -Requirements Process Or Requirements Engineering Process -Requirements Elicitation -Requirements Analysis -Requirements Specification -Requirements Validation -Practical Considerations -Initiation And Scope Definition -Software Requirements Tools	-Software Requirements Fundamentals Or Fundamentals Of Requirements Engineering -Requirements Process Or Requirements Engineering Process -Requirements Elicitation -Requirements Analysis -Requirements Specification -Requirements Validation -Practical Considerations
Software Design	Software Design	-Software Design Fundamentals -Key Issues In Software Design -Software Structure And Architecture -User Interface Design -Software Design Quality Analysis And Evaluation -Software Design Notations -Software Design Strategies And Methods -Software Design Tools	-Software Design Fundamentals -Key Issues In Software Design -Software Structure And Architecture -Software Design Quality Analysis And Evaluation -Software Design Notations -Software Design Strategies And Methods
Software Construction	Software Construction	-Software Construction Fundamentals -Managing Construction -Practical Considerations -Construction Technologies -Construction Tools	-Software Construction Fundamentals -Managing Construction -Practical Considerations
Software Testing	Testing	-Software Testing Fundamentals Or Testing Fundamentals -Test Levels -Test Techniques -Test-Related Measures -Test Process -Software Testing Tools	-Software Testing Fundamentals Or Testing Fundamentals -Test Levels -Test Techniques -Test-Related Measures -Test Process
Software Maintenance	Software Maintenance	-Software Maintenance Fundamentals -Key Issues In Software Maintenance -Maintenance Process -Techniques For Maintenance -Software Maintenance Tools	-Software Maintenance Fundamentals -Key Issues In Software Maintenance -Maintenance Process -Techniques For Maintenance
Software Configuration Management	Configuration Management	-Management Of The Scm Process Or Management Of The Cm Process -Software Configuration Identification Or Configuration Identification -Software Configuration Control Or Configuration Control -Software Configuration Status Accounting Or Configuration Status Accounting -Software Configuration Auditing -Software Release Management And Delivery -Software Configuration Management Tools.	-Management Of The Scm Process Or -Management Of The Cm Process -Software Configuration Identification Or Configuration Identification -Software Configuration Control Or Configuration Control -Software Configuration Status -Accounting Or Configuration Status Accounting -Software Release Management And Delivery
Software Engineering Management	Software Engineering Management	-Initiation And Scope Definition -Software Project Planning -Review And Evaluation -Closure	-Software Project Planning -Software Project Enactment -Review And Evaluation -Closure

		-Software Engineering Measurement -Software Management Tools -Risk Management -Software Project Organization And Enactment -Engineering Economics	-Software Engineering Measurement -Software Project Enactment
Software Engineering Process	Software Engineering Process	-Process Implementation And Change -Software Process Definition Or Process Definition -Software Life Cycles -Software Process Assessment And Improvements -Software Engineering Process Tools - Product And Process Measurement Or Measurement	-Process Definition -Process Assessment -Measurement Or Product And Process Measurement
Software Quality	Software Quality	-Software Quality Fundamentals -Software Quality Management Processes -Software Quality Practical Considerations -Software Quality Tools	-Software Quality Fundamentals -Software Quality Management Processes
Software Engineering Professional Practice	Ethics And Professional Conduct	-Professionalism -Group Dynamics / Psychology -Communications Skills -Social, Legal, And Historical Issues -Codes Of Ethics And Professional Conduct* -The Nature And Role Of Software Engineering Standards	
Software Engineering Economics		-Software Engineering Economics Fundamentals -Life Cycle Economics - Risk, And Uncertainty -Economics Analysis Methods -Practical Considerations	
Computing Foundations	Computing Fundamentals	-Computer Science Foundations -Programming Fundamentals -Data Structures And Algorithms -Computer Architecture -Operating Systems -Networks And Communications -Module Design And Construction	
Mathematical Foundations	Mathematics Fundamentals	-Mathematical Foundations -Discrete Structures -Propositional And Predicate Logic -Probability And Statistics	
Engineering Foundations		-Empirical Methods And Experimental Techniques -Statistical Analysis -Measurement -Engineering Design -Modeling, Prototyping And Simulation -Standards -Root Cause Analysis	
	Software Engineering	-Software Requirements -Software Design -Software Construction -Software Testing -Software Maintenance -Software Engineering Management -Software Engineering Process -Software Quality	

Appendix 2: SWEBOK Guide and CBOK difference at subarea level

SWEBOK SUBARAES	CBOK SUBARAES	SWEBOK – CBOK (S-C) SUBARAES	CBOK –SWEBOK (C-S) SUBARAES
Software Requirements	Requirements Engineering	-Software Requirements Tools	-Initiation And Scope Definition.
Software Design	Software Design	-User Interface Design -Software Design Tools	
Software Construction	Software Construction	-Construction Technologies -Construction Tools.	
Software Testing	Testing	-Software Testing Tools	
Software Maintenance	Software Maintenance	-Software Maintenance Tools	
Software Configuration Management	Configuration Management	-Software Configuration Auditing. -Software Configuration Management Tools	
Software	Software	-Initiation And Scope Definition.	- Risk Management.

Engineering Management	Engineering Management	-Software Management Tools.	- Engineering Economics. -Software Project Organization (Part Of Subarea).
Software Engineering Process	Software Engineering Process	-Software Life Cycles -Software Process Improvement (Part Of Subarea). -Software Engineering Process Tools.	-Process Implementation And Change
Software Quality	Software Quality	-Software Quality Practical Considerations. -Software Quality Tools	-Verification And Validation (V&V)
Software Engineering Professional Practice	Ethics And Professional Conduct	-Group Dynamics / Psychology -Communications Skills - Professionalism	-Social, Legal, And Historical Issues. -The Nature Role Of Software Engineering Standards. -Codes Of Ethics And Professional Conduct