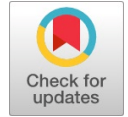# Validating the Knowledge Acquisition Process Metrics in Content Management Systems

Jason Jeyasingh, Justus S

Abstract: *Measuring the processes involved in knowledge engineering for designing and building an intelligent system has taken significant role. Out of the four basic processes involved in knowledge engineering, this paper deals with the knowledge acquisition process and the metrics necessary for measuring the process itself. Three metrics are proposed for the knowledge acquisition process based on the entailment procedures, its length and complexity, and the cohesion and coupling attributes of the collection of knowledge units. These three metrics are formalized based on the Briand's mathematical properties for validating software metrics. These metrics are indicative in the way it gives an insight on the design and the development of a knowledgebase. In addition to these metrics, newer metrics can also be proposed for knowledge representation and knowledge sharing processes.*

Index Terms: *Knowledge Engineering, Knowledge Acquisition, Software Metrics, Metrics validation.*

I. INTRODUCTION

There are several metrics frameworks and models defined for software systems since the 1990s. Some of them are traditional metrics like lines of code (LOC), functional point (FP), object-oriented metrics like object-size (OS), object points (OP), cohesion, coupling, number of children (NoC) etc., and object-relational metrics like table size (TS), runtime cohesion and runtime coupling, number of inherited properties (NIP) and many more (Justus 2011). However, software metrics are broadly classified into product metrics, process metrics and project metrics. While project metrics and process metrics are mostly indirect and deal with the non-functional attributes, the product metrics deal with the structural and functional aspects of the product. The structural metrics are collected during the design time of the product and the functional metrics are collected during the runtime (Calero 2001, Chidembar 1994).

A. Motivation

There are lots of metrics for Software Systems, which range for system design metrics (Manik Sharma2011; Ming 2013), product metrics(Kunal 2015; (Prabhjot 2016), object oriented metrics(Chidembar 1994; Rani 2014), process metrics, andmetrics even for knowledge management(Daniela Et. Al. 2018; Tulawat 2012).But these metrics cannot be applied for a knowledgebase design and functionality. Though some of the object-oriented metrics

like cohesion and coupling (Rani 2014)may well apply to knowledgebase design and operations require additional metrics that may indicate the performance of its knowledge acquisition processes. Predictive metrics for knowledge bases play a vital role in developing a better knowledgebase for our expert systems.

B. Outline

Section 2 details on the overview of the knowledge engineering processes. Section 3 presents the software metrics for knowledge acquisition along with definition, explanation and interpretation of the metrics. Section 4 presents the validation of these metrics based on Briand's mathematical properties to validate software metrics, and section 5 gives the directions for further works in this field of study.

II. OVERVIEW OF KNOWLEDGE ENGINEERING PROCESSES

Knowledge Engineering (KE) deals with the processes and activities in engineering the design of a knowledge based or knowledge driven intelligent smart systems. Knowledge Managements (KM) is all about the processes and tasks at the process-level in an organization, that is, knowledge based organizational learning.

A. Process flow in Knowledge Systems

Knowledge Management itself is a concept or framework that is highly complex, ambiguous, controversial, and difficult to be covered in its entirety. Models and frameworks for these processes would enable a definitive state for KM. The flow of the KM processes is given in figure 1.
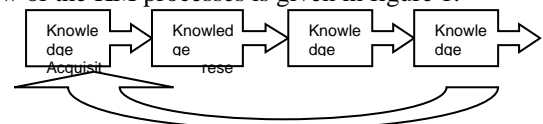

Fig. 1: Stages In Knowledge Management

These stages in KM are sequential in nature, and it begins with interacting with the external system in acquiring knowledge as the first step. The acquired knowledge is then represented in a required format based on the ontology of the system and later deemed for sharing (Christophe 2010) The stored knowledge of a domain is reused by the same KMS or an external KMS, both resulting in derivation of new hybrid knowledge which is added to the knowledgebase. These four main stages of KM cover a wide range of engineered activities, as the knowledge flows into each of these stages (Hepsiba 2016). The different forms of knowledge are handled at each stage in order to be codified, stored and retrieved, so that the knowledge bases serve as an essential backend for the intelligent knowledge based systems.

B. Knowledge Acquisition Processes

Out of the four signification knowledge engineering processes, knowledge acquisition plays a very prominent role. Acquisition of knowledge has to happen from the external world which has structured, semi-structured and unstructured repositories from which the required knowledge is acquired. These repositories are not limited but have open scope for expert systems (ES) to populate their KB with new or hybrid knowledge.

a) Structured Repositories

The most obvious structured repository is relational database management systems (RDBMS). Traditionally these are repositories which have complete control over its data, have more rules about the data, and cater to a specific structure of the data, with the aim of creating a super dataset for its software applications (KorcanKavusan 2016). As the database schema is a collection of tables that are the codifications of the entity, acquiring the clause for FOL representation is easier. Processing the metadata of the schema gives the basic collection of concepts which are then entailed to the KB (Martin 2017). The main advantage of this curated RDBMS structure is that they can be transformed to a concept and KU with less computational power. Some more examples are date represented in XML, OWL, and other semantic structures of web. But not all data are structure.

b) Semi-Structured Repositories

This category of repositories is also explicit in nature but have different formats of data with no specific rules or semantics on the structure of data. A news article on a newspaper, a newsfeed from a webpage, a blog or post in a social networking site, voluminous books and chapters are typical example of semi-structured data. Though they are explicit in nature, they don't have a format, and there's where a lots of knowledge acquisition algorithms demand high research (Briand 1996). Processing power of these repositories is very high involving so many ETs and procedures which may or may not result in a new KU.

c) Un-Structured Repositories

This type of repositories is used in every sort of applications of mobile apps, and they are exponentially growing in its size, usage, and the rate of creation of the KUs is also equally large. Unstructured repositories include images, audio files, video files, streaming data, and handwritten scripts. These are obviously available in social networking sites, servers handling live streaming data, medical transcripts, etc. These are in the explicit form of knowledge, processing these repositories are costly and need efficient algorithms and procedures and extract knowledge (Korcanet. al. 2016). Transformation of implicit knowledge present in this unstructured form to explicit form of representable knowledge demands well formalized knowledge acquisition techniques (Hepsiba 2016).

C. Representation of Acquired Knowledge

Knowledge Discovery techniques, RNN approach, Text analysis and Natural Language processing techniques are followed to acquire knowledge from all of the above mentioned repositories. However, we followed a combination of one or more of these techniques and the resulting knowledge unit is represented as a triplet.

Consider K as the collection of newly acquired knowledge. K = {k1, k2, k3,..., kn} and ki = <e1, T, R, e2>

Where e1 and e2 and newly disscovered entities, T is the type of the entity, R is the relation that exists between e1 and e2.

This triplet of a knowledge unit is then represented in terms of knowledge graph of conceptual graph, frames and defined using description logic.

The knowledge graph representation of the acquired knowledge is more static. It is an acquisition process where the process is done once and that constructs the knowledge graph. However, knowledge acquisition is dynamic and we need to have a more dynamic knowledge representation structure which would acquire knowledge and grow from time to time.

Conceptual Graph (CG) on the other hand is dynamic representation structure that is more suitable in manipulating knowledge units. Conceptual graph, according to John F. Sowa, (John 2001) is "A finite, connected, undirected, bipartite graph with nodes of one type called concepts and nodes of the other type". Conceptual graphs address in terms of concepts and its attributes. A concept can be an entity, event or an action. Every concept has its own attributes and is instantiated with instances. The formation of conceptual graphs is as follows:

• Every arc a of G is a pair (r, c) consisting of a conceptual relation r and a concept c in G. The arc a is said to belong to r; if it has link r to c; but it does not belong to c.

• A conceptual graph G may have concepts that are not linked to any conceptual relation; but every arc that belongs to any conceptual relation r in G must link r to exactly one concept c in G

The metrics proposed in this work are in alignementwith the acquisition processes and the entail processes that makes knowledgebases more compatible with the Expert Systems.

III. METRICS DEFINITION

This section gives the definition of metrics for knowledge processes. Though there are several processes and sub-processes in a knowledge engineering activity, we have taken only one basic process for this metric proposal. The other processes like knowledge representation, knowledge codification and knowledge transfer will be dealt in future research.

A) Knowledge Acquisition

A knowledge unit is acquired through various methods – learning, logical entailments, sharing etc. In this work we will focus on the entailment procedures and the metrics that can be collected out of it.

Logical entailments can be done either using forward chaining or backward chaining approach (Briand 1996; Ronald 2004). Both of these approaches uses clauses from the real-world are taken as its input. For example consider the statements: Sky is clear and blue and Clear skies don't bring rain. The clauses identified in these statements are sky, clear, blue, rain. The relevant clauses are Umbrella, flood, wind etc.

Using these clauses we could logically entail '・Rain・・Umbrella'.

In order to arrive at this conclusion, a forward chaining entailment procedure is run which makes use of the clauses that are already available in the KB. The entailment procedure works out on an entailment tree (ET) which spans until a NULL clause is obtained. Moreover, for an intelligent system, several parallel entailment algorithms would run leading to several ETs, and each KU derived out of these entailments are added or updated in the KB. This is the knowledge acquisition process using logical entailment procedures.

Objective: To evaluate the effectiveness of the knowledge acquisition process

Name of the Entity: Knowledge Acquisition

Type of Entity: Process

Attribute of Measure: Effectiveness

Proposed Metrics: Entailment Depth (ED), Rate of Knowledge Acquisition (RKA), Volume of Knowledge Acquisition (VKA)

Type of Metrics: Direct Measure

With this objective we proceed to propose these metrics in the following sub-sections.

B) Metric: Entailment Depth (ED)

The logical entailment procedure using forward chaining constructs of one or more entailment trees (ET). Getting a NULL clause at the bottom of the ET means, the introduced clause at the above level is valid. Hence the depth of ET is a direct measure of the levels of entailment it took to derive the NULL clause.

ED = The depth of the ET is the entailment level of the tree at which the NULL clause if derived.
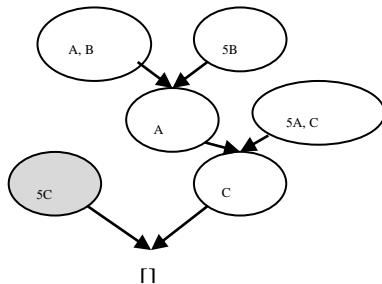

Fig. 2: Example Of An Entailment Tree

Figure 2 gives an example of ET. (A,B), (・B), (・A,C), are the concepts already available in the KB, and they are validated during previous entailment derivation. (A) and (C) are intermediate concepts, but need not be a valid one. (・C) is the new concept that is introduced in the entailment procedure which led to a NULL set. According to entailment algorithm, (・C) is newly accepted concept that can be added to the KB. Any new concept can be introduced at any level in this derivation, but the condition is that it should lead to a NULL set.

In this example, the entailment depth (ED) is 4. Regardless of at which level the new concept is introduced, the depth of the ET gives an insight into how large is the entailment computation.

Entailment Depth value	Indication
Low	▪ Less concepts are involved in the entailment procedure ▪ Less computation time required ▪ Instant new knowledge acquisition
High	▪ Less concepts are involved in the entailment procedure ▪ Less computation time required

Table. 1: ED Metric Value Indications

ED indicates the no of entailment levels involved in the chaining procedures. ED and the number of clauses involved in the entailment procedure are directly proportional. The proportionality constant is the number of variable assignments to the entailments at each level. ED also indicates the computational complexity involved in the executing the entailment tree.

C) METRIC: RATE OF KNOWLEDGE ACQUISITION (RKA)

Every entailment procedure or the entailment algorithm execution may or may not lead to a new knowledge concept. In most cases, the ET ends up in a NULL and the new knowledge introduced in the ET will be added to the KB. However, in some cases, if the ET ends up in a clause, that is, a Not-NULL clause, and there is not a clause to compliment it, then the ET doesn't resolve to form a new knowledge concept. In the former case, the rate at which new knowledge concepts are acquired from the ET can be computed. The definition of RKA is given as:

$$RKA = \frac{\text{the number of new knowledge units acquired}}{\text{total number of unique concepts in the entailment tree}}$$

The RKA for a single ET is the number of new knowledge units validated and added to KB over the total number of unique concepts involved in the entailment procedure. The average RKA will be the average of the RKAs collected over a period of time. The RKA for the example shown in figure 4 is calculated to be 0.2 KUs per ET. The interpretation for the RKA metric is given in table 2.

Rate of Knowledge Acquisition (RKA) value	Indication
Low	▪ Less new knowledge acquired ▪ Number of KUs used for entailment is high ▪ Appropriateness of the new KUs might be less
High	▪ More knowledge acquired in a given time in a given set of entailment trees ▪ The frequency of KUs added to KB is high ▪ KBs growth in terms of KU acquisition is good.

Table. 2: RKA metric value indications

We considered this passage taken from The Hindu online edition (The Hindu 2017).

Fig. 3: Sample Text For Processing Using ETS

Fig. 4: Acquired KUS Using ETS

Figure 3 is the input text for the acquisition algorithm which uses forward chaining entailment technique sections A-D below for more information on proofreading, spelling and grammar.

Total Words in the Text	Total Stop words	KUs acquired	Total ETs	RKA
103	46	15	9	1.78

Table. 3: KRA Metric Values

Figure 4 shows the boxed texts as the KUs acquired, and the shaded boxes are the related KUs. Table 3 gives the metric value for knowledge acquisition for the text shown in Figure 3.

RKA can be calculated for a single ET and also for a set of ETs. Estimate of single ET, and then sum the values to get the metric value of the entailment procedure for a KB in a given time duration. RKA gives an indication to the velocity aspect of Bigdata (Francis 2016). It is also an indirect metric which indicates the complexity of the entailment procedure involved in forming a new concept.

D) Metric: Volume of acquired knowledge (VAK)

While the rate of knowledge units acquired indicates velocity, the volume of the acquired KUs directly indicates the volume aspect of Bigdata(Francis 2016). A knowledgebase will be a repository of bigdata as it learns and entails more KUs over time.

Processing a text for knowledge acquisition involves many entailment trees, and this one processing is called entailment procedure (EP). In one EP, with more than one ETs, we may acquire new KU or an existing KU be updated or a new allied, hybrid KU. The total sum of all these KUs acquired per EP is the volume of KUs acquired. The definition is as follows:

$$VAK= \frac{No.of\ new\ knowledge + No.of\ hybrid\ knowledge + No.of\ knowledge\ updates}{Entailment\ Procedure}$$

For the example given in Figure 4, the newly acquired KUs are 15. One of the new KU here is water. However, this is not directly relevant to this text. Hence in a later EP if this KU is entailed to be 'water resources', then this is an updated KU.

In figure 5, we have a new KU, · rainfall, from ET1 and a composite hybrid KU, [·drought, monsoon], from ET2. This KU from ETS is normalized to (drought · · monsoon), which is interpreted as 'if drought then there is no monsoon'. This interpretation satisfies the KB because we have a NULL at the end of the derivation.

No. of ETs (a)	No. of concepts involved (b)	No. of new KU (c)	No. of updated KU (d)	No. of hybrid KU (e)	RKA	VAK = ((c+d+e)/a)
2	11	1	1	1	0.458	1.5

Table. 4: Calculating The Metrics For Knowledge Acquisition

VAK value	Indication
Low	▪ The total KU acquisition is less for an entailment ▪ Less or no KU is acquired. Hence either the entailment concepts taken might be irrelevant, or no sufficient KUs are available with the KB. It will take time and iterations for the KB to acquire more basic concepts
High	▪ KU acquisition to the KB is good ▪ KB is already having sufficient concepts that are relevantly used in entailment procedures.

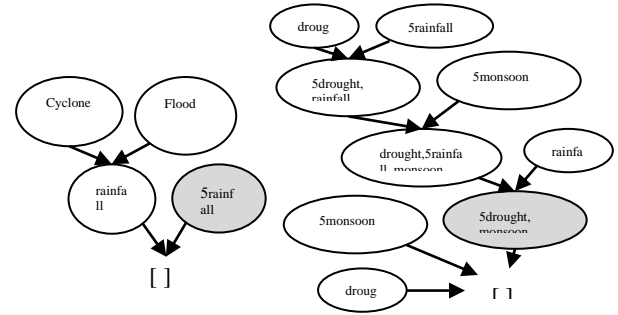**Table. 5:** VAK Metric Value Indications

Fig. 5: Entailments Derives to New KU and Hybrid KU

New KU: · rainfall

Hybrid KU: [·drought, monsoon]

Applying First Order Logic conversion

(drought · · monsoon)

Table 4 shows the metrics that are directly calculated from the ET and its entailment procedures. Table 5 shows the interpretation of the VAK metric with respect to the amount of different types of KUs being entailed and added to the KB for every entailment procedure.

IV. FORMAL VALIDATIONS OF THE METRICS

The proposed three metrics Entailment Depth (ED), Rate of Knowledge Acquisition (RAK), and Volume of Acquired Knowledge (VAK) are validated based on the Briand's mathematical properties for software metrics validation.

In (Briand 1996), the authors proposed a set of mathematical properties and important internal software attributes for "size, length, complexity, cohesion, and coupling" measurement that are formalized. Their approach of validation is perceived as a set of metric

A system S is defined as a pair of <E, R>, where E represents the set of elements of S, and R is the binary relation on E (R · E*E), representing the relations between the metric values in E. Given a system μ = <Em, Rm> is a metric of S, and only if Em · E, Rm · Em*Em, Rm · R.

Now, the metric system MS = <E, R, μ> represents S and μ is the set of metric values for S.

A) Entailment Depth (ED): Metric Validation

This metric is about the length of the entailment tree involved in finding new knowledge. It is obvious that the property length can be used to validate this metric.

Non-Negativity: The system S is the entailment tree which is the subject of measure. S=<E,R> is defined as E being the set of knowledge units(KU) and R being the entailment operation on the KUs. The length of S is non-negative, ie., Length(S) > 0. Hence the μ=<Em,Rm> is also non-negative.

Null Value: The length of the system S is null, if E is empty. Hence, μ = null, if E is empty, ie., no KUs are available for entailment.
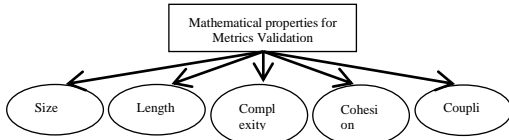
260

Fig. 6: Attributes To Validate the Metrics

Non-increasing Monotonicity of Connected Components: Let S be an entailment tree and m be a sub-tree of S, such that m is represented as the connected components of tree S.

Adding entailment relation between m and S does not increase the Length of S. (S=<E,R>, and m = <Em, Rm> and m · S, and m· is a connected relation of S · , where S · = <E, R · > and R · = {R · <e1,e2>} and <e1,e2>· R, and e1 · Em1 and e2 · Em2), hence Length (S) · Length (S ·).

Non-decreasing Monotonicity of Non-connected Components: Let S be an entailment tree, m1 and m2 be a sub-trees of S, such that m1 and m2 are two separate connected entailment tress. Adding a conceptual relation from elements of m1 to elements of m2, does not decrease the length of S. (S=<E,R>, and m1 = <Em1, Rm1> and m1 · S · and m2 · S · ·. R' = R _ {<e1, e2>} and <e1,e2>· R and e1 · Em1 and e2 · Em2), implies Length(S') · Length(S).

Disjointness: Let S = <E, R> is an entailment tree comprised two disjoint sub-trees m1 and m2. The total length of S is the maximum of the lengths of m1 and m2. Since S = m1· m2, and m1· m2 = · , Length(S) = max[length(m1), length(m2)].

B) Rate of Knowledge acquisition (RKA): Metric Validation

This metric deals with the rate at which new knowledge units are entailed, and stored in the Knowledgebase. The attribute Complexity is used to validate this metric RKA, because it is an indirect metric to assess the complexity of the entailment procedure. The system S is an entailment procedure that delivers new knowledge units to the knowledgebase. Hence, the complexity attributes to validate RKA metric.

Non-Negativity: Complexity of S = <E, R> is always non-negative. Either new knowledge is created through an entailment procedure or not, the complexity of S is greater than zero. The complexity of S is non-negative, ie., Complexity(S) > 0. Hence the μ=<Eep,Rep> is also non-negative.

Null Value: The metric RKA of the system S is null, if E procedure is not initiated or becomes invalid due to execution faults. Hence, μ = null, if E is not executed.

Non-increasing Monotonicity of Connected Components: Let S be an entailment procedure and m be a sub-tree of S, such that m is represented as the connected components of tree S. Adding 'the complexity in acquiring knowledge units' relation between m and S does not increase the RKA value of S. (S=<E,R>, and m = <Em, Rm> and m · S, and m· is a connected relation of S · , where S · = <E, R · > and R · = {R · <e1,e2>} and <e1,e2>· R, and e1 · Em1 and e2 · Em2), hence RKA (S) · RKA (S ·).

Non-decreasing Monotonicity of Non-connected Components: Let S be an entailment procedure, m1 and m2 be a sub-trees of S, such that m1 and m2 are two separate connected entailment procedures. Adding a complexity relation from elements of m1 to elements of m2, does not decrease the RKA value of S. (S=<E,R>, and m1 = <Em1, Rm1> and m1 · S · and m2 · S · ·. R' = R _ {<e1, e2>} and <e1,e2>· R and e1 · Em1 and e2 · Em2), implies RKA(S') · RKA(S).

Disjointness: Let S = <E, R> is an entailment procedure comprised two disjoint sub-procedures m1 and m2. The total RKA value of S is the maximum of the RKA values of m1 and m2. Since S = m1 · m2, and m1 · m2 = · , RKA(S) = max[RKA(m1), RKA(m2)].

C) Volume of acquired knowledge (VAK): Metric Validation

The knowledge units are added to the knowledgebase (KB) can be in the forms of new knowledge, updation on the existing knowledge or hybrid knowledge from the existing knowledge. Hence we will validate this metric with the Size attribute. However, this collection of knowledge units that are added into the KB are cohesive within themselves and coupled with other knowledge units (KU). Hence, the validation of the metric partly involves cohesion and coupling of these KUs.

S = <E, R>, where E is the new collection of KUs added to the KB as a result of an entailment procedure, and R is the relation of cohesiveness or coupling acting on the collection E.

Non-Negativity: Size of S = <E, R> is always non-negative. Either a new knowledge is created or an existing KU is updated, the size of S is always greater than zero. Hence the μ=<Ev,Rv> is also non-negative.

Null Value: The size of the system S is null, if no new KUs are added or updations have not happened or becomes invalid due to execution faults. Hence, μ = null, if E is unaltered.

Non-increasing Monotonicity of Connected Components: Let R be the cohesiveness of S and Rm be the cohesiveness of a sub-tree of S, such that m is represented as the togetherness of S. Adding the cohesion of R and Rm of S does not increase the VAK value of S. (S=<E,R>, and m = <Em, Rm> and m · S, and m· is a connected relation of S · , where S · = <E, R · > and R · = {R · <e1,e2>} and <e1,e2>· R, and e1 · Em1 and e2 · Em2), hence VAK (S) · VAK (S ·).

Non-decreasing Monotonicity of Non-connected Components: Let R be the connectedness (coupling) of S, and Rm1 and Rm2 be the connectedness of S, such that m1 and m2 are two separate coupling values. Adding the coupling values of the relation R from elements of m1 and elements of m2, does not decrease the connectedness of VAK of S. (S=<E, R>, and m1 = <Em1, Rm1> and m1 · S· and m2 · S · ·. R' = R _ {<e1, e2>} and <e1,e2>· R and e1 · Em1 and e2 · Em2), implies VAK(S') · VAK(S).

Disjointness: Let S = <E, R> is the KUs to be added or updated in KB through an entailment procedure comprised two disjoint sub-procedures m1 and m2.

The total VAK value of S is the maximum of the VAK values of m1 and m2. Since S = m1 · m2, and m1 · m2 = · , VAK(S) = max[VAK(m1), VAK(m2)].

V. CONCLUSION & FUTURE DIRECTIONS

This work on proposing the definition of three metrics for knowledge acquisition process, and their validations based on the Briand's mathematical properties(Briand 1996) to validate metrics have formed the basis for building a knowledgebase. Out of the four processes in knowledge engineering, the first process, knowledge acquisition is considered in this paper, and possible metrics, their definition, explanation, the motivation of the metric, the metric collected values and their interpretation are well defined. Moreover, these metrics are validated on the basis of the five properties of metrics validation(Srinivasan 2014).

In addition to these metrics, newer metrics can be proposed for knowledge representation, knowledge coding and sharing. They have wide scope in the design and development of knowledgebase for expert system. Intelligent knowledgebases are replacing the traditional relation, NoSQL structures. The plain representation formats of MANGODB are widely used, though with their own limitations. When knowledgebases can be built based on object-relational data modelling and databases, they will become the right choice for intelligent systems. Hence measuring the knowledge processes lead to better design of knowledgebase.

REFERENCES

1. Arisha, A. and Ragab, M. (2013).The MinK Framework: Developing Metrics for the Measurement of Individual Knowledge. KIM2013 Knowledge & Information Management Conference, UK, 4 – 5 June, 2013.
2. Briand, L.C., Morasca, S., Basili, V.R., (1996) Property-Based Software Engineering Measurement. IEEE Transactions on Software Engineering, 22(1): 68-85.
3. Calero C., Piattini M., Genero M. (2001) Defining Complexity Metrics for Object-Relational Databases. In Patel D., Choudhury I., Patel S., de Cesare S. [eds], Object Oriented Information System. London: Springer
4. Chidamber, S.R., and Kemerer, C.F. (1994)A Metrics Suite for Object-Oriented Design IEEE Transactions on Software Engineering 20(6): 476-493.
5. Christophe F, Bernard A, Coatanéa E. (2010) RFBS: A Model for Knowledge Representation of Conceptual Design. CIRP Annals–Manufacturing Technology, 59(1): 155–158.
6. Churcher, N. I., and M. J. Shepperd.(1995). Towards a Conceptual Framework for Object-Oriented Metrics.ACM Software Engineering Notes. 20(2): 69–76.
7. Daniela Geanina Luca Cososchi, Alina Luca, LuminiţaMihaelaLupu, IonuţViorelHerghiligiu. (2018). Indicators System For Assessing The Organizational Knowledge Acquisition Process. Environmental Engineering and Management Journal, 17(4): 937-950
8. Francis Rousseaux, Stéphane Cormier, (2016) Knowledge Acquisition at the Time of Big Data. Federated Conference on Computer Science and Information Systems, September 11-14 (Poland: FedCSIS), 1343-1348.
9. Hepsiba Mabel V, Justus Selwyn. (2016) A Review on the Knowledge Representation Models and its Implications. International Journal of Information Technology and Computer Science 8(10): 72-81.
10. John F. Sowa, 2001. Conceptual Graphs, draft proposed ISO standard for conceptual graphs. http://www.jfsowa.com/cgcgstand.htm.
11. Justus. S, Iyakutti. K. (2011) An Empirical Validation of the Suite of Metrics for Object-Relational Data Modelling. International Journal of Intelligent Information and Database Systems 5(1): 49-80.
12. KorcanKavusan, Niels G. Noorderhaven and Geert Duysters. (2016) Knowledge acquisition and complementary specialization in alliances: The impact of technological overlap and alliance experience. Research Policy, 45(10): 2153-2165.
13. Kunal Chopra, Monika Sachdeva. 2015. Evaluation Of Software Metrics For Software Projects. International Journal of Computers & Technology.14(6).
14. Manik Sharma, Gurvinder Singh. 2011. Predictive Metric- A Comparative Study. International Journal of Computer Science and Technology (IJCST). 2 (1)
15. Martin Lorenz et. al. (2017) Object-Relational Mapping Revisited - A Quantitative Study on the Impact of Database Technology on O/R Mapping Strategies. International Conference on System Sciences.
16. Maryam Dehghani, PeymanAkhavan, (2017)An experimental investigation of knowledge acquisition techniques. Journal of Management Development, 36(4):493-514.
17. Ming Chang Lee, To Chang. 2013. Software Measurement and Software Metrics in Software Quality. International Journal of Software Engineering and Its Applications 7, (4).
18. Ohwada, Hayato, Yoshida, Kenichi, (2016) Knowledge Management and Acquisition for Intelligent Systems. Lecture Notes in Artificial Intelligence. 14th Pacific Rim Knowledge Acquisition Workshop. August 22-23 (Springer).
19. Prabhjot Kaur. (2016). A Review of Software Metric and Measurement.International Journal of Computer Applications & Information Technology.9(2).
20. Rani Geetika, Paramvir Singh. 2014.Dynamic Coupling Metrics for Object Oriented Software Systems- A Survey. ACM SIGSOFT Software Engineering Notes. 39 (2)
21. Srinivasan. K.P., T. Devi. (2014) Software Metrics Validation Methodologies in Software Engineering. International Journal of Software Engineering & Applications, 5(6): 87-102
22. Ronald J. B, Hector J. L. (2004) Knowledge Representation and Reasoning. (San Francisco: Morgan Kaufmann Publishers)
23. Tse-Hsun Chen et. al. (2016) An Empirical Study on the Practice of Maintaining Object-Relational Mapping Code in Java Systems. 13th International Conference on Mining Software Repositories, May, 2016.
24. Tullawat P &VichitaVathanophas R (2012). Knowledge acquisition: the roles of perceived value of knowledge content and source. Journal of Knowledge Management 16, 724-739.
25. The Hindu, the online daily newspaper, (http://www.thehindu.com/news/national/tamil-nadu/tamil-nadu-where-climate-paradoxes-are-becoming-the-norm/article17530209.ece), visited in August 2017.

AUTHORS PROFILE

Dr. S. Justus received his doctorate degree from Madurai Kamaraj University, Madurai, India. His research specializations include Object-relational data modeling, knowledge engineering and Big Data.

He has been into academic research and has published several of his research work results in International Journals and Conferences – including SwSTE in Israel and DASMA in Germany. He has also practitioner's experience while working with Software development companies.

He is a member of IEEE, ISTE, IAENG professional associations. He has served as research & project coordinator for PG studies at Engineering Institutes. Presently he is working as Associate Professor at VIT University, Chennai.