Design of Knowledgebase for EES using Object - Relational Data Modeling

Justus Selwyn
Department of Computer Science, John Brown University
Siloam Springs 72761, AR, USA

ABSTRACT
This work is focused on designing a knowledgebase for an Expert Education System (EES) that will help the learners to know where, what, how, and when is the required knowledge available in the knowledgebase. A three-layered architecture of a Knowledgebase is proposed, and have given some basic definitions for the concepts, knowledge units, and persistent objects. Definitions are given in FOL, frames, DL, and OR data models. Integrating these into a KB will serve as a backend for the EES.

General Terms
Knowledgebase, Knowledge Engineering, Expert System Design

Keywords
Component; first-order- logic, frames, description-logic, knowledge-units, storage-unit

1. INTRODUCTION
Knowledge is true information which is defined as a "system known true beliefs" [5]. It exists in so many forms - structured, unstructured, and semi-structured forms, [7] and is found available in the repositories. It is found in the human mind, in the real-world around us, in structured database forms and tables, in unstructured audio, video, and images, in semi-structured posts, tweets, short messages on social networking sites, and in news feeds onweb portals.

These forms of knowledge are broadly categorized into two types: Tacit Knowledge and Explicit Knowledge [3]. Tacit knowledge is one that cannot be expressed by logic or language and is in the human mind. However, they can be identified and extracted, and represented in an Explicit form. The knowledge that can be represented in one or more forms is called Explicit knowledge. All the structured, unstructured, and semi-structured forms found in repositories can be transformed or converted to an explicit form of knowledge [4].

1.1 What is Knowledge representation?
Knowledge is everywhere, and possession of knowledge is power. Hence knowledge workers are interested and motivated to identify the existence of knowledge, in their relevant domain, acquire them and represent them to claim its procession [6], [8]. Be it individual, or team, or organization, Knowledgeable is incredible. Only through a representable form of knowledge can it be possessed.

The three steps in explicating knowledge are:

1. Identify and extract the concepts and facts from the repositories

2. Adding rules, associations, and relations to the concepts converts it into a semantic and pragmatic knowledge unit

3. The represented knowledge unit is then codified and stored as a persistent object

To represent an acquired concept into an explicit form of knowledge the three rules basically syntax, semantics, and pragmatics [5] has to hold good. By using knowledge reasoning decision-making is possible from the acquired knowledge.

1.2 Motivation
While the learning process follows conventional and contemporary methods, interactive learning makes the learner to learn fast and acquire the subject knowledge quickly [11]. To make this possible, the learner has to know where to find what, how, and when. Answering these questions from the knowledge worker's perspective has become the motivation to design a knowledgebase that will help the learners to find ways to know which subject's lecture notes or video tutorial is available where and with whom, thus providing the learner with possible sources of study materials.

In this paper, an attemptis made to give a design architecture of a knowledgebase for an "Education Expert System" (EES). Some processes in our University Education System and formulated concepts are identified and converted them to knowledge units, thus enabling knowledge reasoning in EES. In section 2 an overview of the EES is discussed, in section 3 the architecture of the KB is presented and the representations are given in section 4, with concluding remarks at section 5

2. OVERVIEW OF EXPERT EDUCATIONAL SYSTEM
Motivated by the teaching-learning processes of our University, a few facets of the systemare researched. This study leads to this research work on designing a knowledge-based Expert Education System (EES). Some of the features of the EES and the KB are foreseen here.

- EES is mainly focused on the teaching and learning processes. All the tasks performed by the teachers and students are captured and given semantics based on the relations these two primary actors share. For example:*Teacher* offers a *Course*, and *Student* registers for that *Course*. Both are related bi-directionally

- This system deals with "Organizational learning process" which is one of the key model of the capability maturity model (CMM), proposed by the Carnegie Melon University, USA

- CMM is the methodology of process learning, to updation of individual after completing the academia. It includes mainly establishing strategic training capability needs, to determine which training needs responsible, implementing technical plan, and establishing training capability.

- Content management system isused here for publishing and editing the desired data. It acts as the central interface between the learner and the EES, enabling the knowledge worker to manage and manipulate the Knowledgebase,
- The main intention of EES is to help the learner or the teacher to find the knowledge expert in a given subject, his/her contents like learning materials in various forms from doc, ppt, xls, txt to audio, video, image, posts, and tweets, and finally finding the status of the learner.

With this overview of EES, the work presents the 3-layered architecture of the knowledgebase and the representation definitions of knowledge units in each of these layers in the next sections.

3. THREE LAYERED ARCHITECTURE OF KNOWLEDGEBASE

Considering the complexities in computations that will happen in an EES, a three-layered architecture of the knowledgebase (KB) for EESis designed. This is to make the flow of concepts and KUs in the KB and in the EES, and help knowledge workers to manage the KB in designing, reasoning, and storing of KUs.

The Knowledgebase is layered as follows:

1) Design Layer – FOL representations of concepts and facts

2) Knowledge Layer – KU representations adding semantic and pragmatic to help reasoning, entailment computations, and resolving new concepts into the KB

3) Storage Layer – KUs are codified in class_types in object-relational data models that will store the KUs as persistent objects.

3.1 Design Layer

The main function of the layer is to define different relative clauses in different combinations which are represented in FOL (First Order Logic). This is a human-understandable layer. Here new clauses can be resolved from the existing knowledge clauses.

The identified concepts of the design layer are:

Teacher, Student,Company, Institution, Department, Course, Content_Type, Content, SearchFor, TeacherOf, StudentOf, CourseOwner, TakesCourse, PrefersCourse, FrequencyOfAccess, WorksWith, SpecializedIn.

Eg : Person :: student, staff, paren

FOL :: □x □y □z person (x) ε [student(y) ϖ staff (z) ϖ person(w)]

3.2 Knowledge layer

The main function of the layer is to define knowledge units for atomic clauses which are defined in the design layer. This can be represented in either frame representation or knowledge description layer. This is a machine-understandable layer. Here the static & dynamic knowledge key units in frame representation must be derived.

The knowledge units for the system are given in table 1.

Table 1. Table captions should be placed above the table

Atomic Concepts	Derived/Dependent Concepts
Person	Student, Staff, Parent
Courses	Subjects, lectures, labs, tutorials
Institution	DeanOf, Coursesoffered, departments
Media	Audio, video, text, documents, images

These represented knowledge units can be defined using frame definitions or description logic definitions. An example of frame is given here.

```
( Student
<: IS A Person
<: name: IF_ADDED{ update<:program><:dept><:institution>}
<:CGPA :IF_NEEDED{ Return(CGPA calculation)}
<:progam :NULL
<:Dept :NULL
<:Institution:NULL
<:Courses: IF NEEDED{ Return(selected courses)}
    Update This <:program> This<:Dept>This<:Institution>
);
```

3.3 Storage Layer

The main function of the layer is to store the knowledge units as object-relational database (ORDB) objects in the storage device. The knowledge units in the knowledge layer along with their relations are defined as a class_type in ORDB, and the schema is designed out of the collection of such other class_types. The constants associated with the concepts and KUs are instantiated as objects of the class_types. The programming platforms are JAVA or PHP to store them by using Object-Relational Database (ORDB), File system.

```
Create  Student_t: extends Serson_t
{
        name String;
        regd_no String;
        programme String;
        cgpa float;
        ContactAddressAddress_t;
Relation:
        StudentOf (this, staff_t);
        BelongsToDepartment (this, department_t);
        RegisteredCourses (this, courses_t);
}
```

Fig 1: Class_Type Definition in Knowledgebase

4. REPRESENTATION OF KUS IN KNOWLEDGEBASE

The knowledgebase is divided into two parts: Part-A and Part-B. These two parts of the KB works in coherence, where query executions, resolutions, entailments, and reason happen in Part-A, with the help of the content in Part-B. This means, the design-layer and the knowledge-layer are in Part-A, and the storage-layer, its schemas, and class_types are in the Part-B of the KB. In this section, the concept representations in the three layers are presented.

4.1 Design Layer Representation

The identified and acquired concepts and facts from the structured and unstructured repositories are represented as concepts and clauses in the first-order-logic (FOL). Four basic atomic clauses/concepts, from which other dependent concepts are derived and/or they function based on the existence of the atomic clauses are focused.

The Atomic clauses identified are: Person, Courses, Institution, Media

The dependent or extended clauses are Student, Staff, Subjects, Lectures, Departments, CoursesOffer, Audio, Video etc.

These existential properties of these clauses are represented as follows:

1) **Person :: student, staff, parent**
□x □y □z Person (x) ε [Student(y) ϖ Staff (z)]

2) **Courses :: subjects, lectures, tutorials, lab**
□x □s □c □t □l Courses (x) ε [Subjects(s) ω Lectures(c) ω Tutorials(t) ω Lab(l)]

3) **Institution :: HOI, courses offer, departments**
□x □v □w □y □z Institution(x) ε Departments(v) εCoursesOffer(w)
[staff(y) ϖ students(z)]

4) **Media :: audio, video, text, Document, images**
□x □y □y' □z □z' subjects(x) ε [Audio(y) ϖ video(y') ϖ text(z)ϖ documents(z') ϖ images(w)]

Fig 2: Definitions using First Order Logic

4.2 Knowledge-Layer Representation

The most important part in knowledge representation is done in this layer. Careful study of the concepts and facts leads to the derivation of associations and rules for the concepts. With these rules and associations, the knowledge units are defined in two representation models, frames and description logic.

The atomic concepts are defined using frames, where the slots are filled using the three-rules IF-ADDED, IF-NEEDED and IF-REMOVED. The other concepts are defined in DL, where the concepts are made dependent or derived concepts from other concepts. The DL representation involves DL operators like AND, ALL, EXISTS, and FILLS. The frame representations are given in Figure 2, and DL representations are given in Figure 3.

(**Student**
<: IS A Person
<: name**: IF_ADDED**{
update<:program><:dept><:institution>}
<:CGPA :IF_NEEDED{ Return(CGPA caluculation)}
<:progam :NULL
<:Dept :NULL
<:Institution:NULL
<:Courses: IF NEEDED{ Return(selected courses)}
Update This <:program>
This<:Dept>This<:Institution>
);

(**Staff**
<: IS A Teacher/person
<:name :IF ADDED{ update<:dept><:degree>}
<:subject :IF NEEDED{Return(subject Allocation)}
<:Department: NULL
<:Degree: NULL
<:Salary : IF_NEEDED {Return(salary caluculation)}
Update This<:dept>This<:Degree>
);

(**Parent**
<:IS A Person
<:name :IF_ADDED {update<:Student name>}
<:Address : IF_NEEDED { Return (Address info)}
<:Contact no : IF_NEEDED { Return (Contact info)}
);

(**Courses**
<:IS A Stream
<:name : IF ADDED { Update
<:subject><:Dept><:Teacher><:duration>}
<:subjects:NULL
<:Departments : NULL
<:Teacher :NULL
<:duration :NULL
Update This <:Dept>
This<:Teacher>This<:duration>
<:Labs: IF NEEDED {Return(Labs)}
);

(**Institution**
<: IS A UNIVERSITY
<: name :IF NEEDED Return(name of the institution)
<:HOI : IF NEEDED Return (name of HOI)
<:CoursesOffered : IF NEEDED Return (name of the courses)
<:Department: IF NEEDED Return (name of the departments)
);

(**Media**
<:IS A Data
<:Type of file : IF ADDED{update<:size>}
<:Teacher : IF ADDED {Update <:name>}
<:size :NULL
<:name: null
<:SUBJECT : IF ADDED {update <:TYPE OF FILES >}
<:Type of files : NULL
Update This <:size >This<:name> This<:Type of file>
);

Fig 3: Frame Representations of KUs

Student = [**AND** Person
[**EXISTS** 1 higher education
[**ALL** Courses [**FILL** Registered]]
[**FILLS** min 6 CGPA]
[**FILLS** min 75% Attendance]
[**FILLS** no due FEE PAID]]

Teacher = [**AND** Person
[**EXISTS** 1 master degree]
[**EXISTS** : Professor [**FILLS** 15years Experience]
[**FILLS** good teaching capabilities]]

Parent = [**AND** : person
[**EXISTS**:min 1 son as a student]
[**FILLS** Attended Parents meeting in college]
[**FILLS** no due FEE PAID]]

Media = [**AND** : Content
[**EXISTS** : DEefferentFormates [**FILLS** : Clear data no noice/errors]]
[**FILLS** : Media [**FILLS** : All course contents]]
[**ALL** : Contents [**FILLS** : Required formats]]]

Subjects = [**AND** : theoretical Subject [**EXISTS** : MIN5 advaced topics] [**ALL** : SUBJECTS [**FILLS** : Teachers are allotted] [**FILLS** : 45 hrs duration]]
Lab = [**AND** : Practical subject [**EXISTS** : min5 Advanced experiments] [**ALL:** experiments [**FILL**: ALL experiment instruments available] [**FILLS** : all are approved by academics]
Tutorial= [**AND** : Reference book [**EXISTS** :all explained topics] [**ALL** : tutorials[**FILLS**: Understands by stuents]] [**FILLS** : Desighned by professors in University]]
Institution = [**AND** : university [**EXISTS** : min 5 Department programms] [**FILLS** : STUDENTS] [**FILLS** : FACULTIES] [**FILLS** : DEPARTMENTS] [**FILLS** : COURSES] [**ALL** : criteria [FILL APPROVED BY govt acts]]
HOI = [**AND** : Person [**EXISTS** : min20 TEACHING experience] [**EXISTS** : min 5years management experience] [**EXISTS** : Doctorate in EDUCATION] [**ALL** :PROGRAMS [**FILL** : Leadership qualities]]
Departments = [**AND** : Section [**EXISTS** : Required courses] [**FILLS** : COURSES [**FILLS** : SUBJECTS] **[ALL** : SUBJECTS [**FILLS** : TEACHERS]]

Fig 4: Description Logic Representations of KUs

4.3 Storage Layer Representations

The knowledge units that are defined in frames and KUs are mainly used to resolving new concepts and KUs, and for computing entailments. However, these KUs need to be stored permanently in a storage media, a secondary memory disk. Hence KUs are codified as class_types of Object-relational data models, and stored in ORDB. These class_types can be instantiated and the constants for the KUs are stored in these objects. Some of the class_type definitions are given in Figure 4.

Create type Person_t
{
String name;
String Department ;
String course;
String institution ;
}
	Create student_t; extends person_t
	{
	String name ;
	String id ;
	String programme;
	Float cgpa ;
	Int semester ;
	Relation:
	StudentOf;
	BelongsToDepartment;
	RegisteredCourses;
	}

Create institution_t
{
String name ;
String address ;
String head_of_institution;
String Courses offer[ ];
String Departments[ ];
 	
Relation:
Institutionanniversary;
Instittutionservices;
}
	Create Course_t
	{
	Sringcoursename;
	String subjects [ ] ;
	Float duration ;
	Intno_of_lectures ;
	Relation:
	Coursewith;
	CourseOf;
	Registeredcourses;
	}

Create media_t
{
String type_of_files [ ];
Int size;
String teacher content ;
String Typeofmedia
String MediaLocation
Relation:
MediaOfCourse();
 	
}
	Create type department _t : extends Teacher _t
	{
	String department name ;
	String department dean ;.
	String courses[ ];
	String Subjects[ ];
	String teachers[ ];
	Relation:
	Department_Hod;
	Deparment_aims;
	}

Create parent : extents person_t
{
String parentname;
String parent_address;
Intcontact_num ;
Relation:
Parent_meetings;
Parent_complaints;
}
	Create type Teacher_t : extends Person_t
	{
	String name ;
	Intemp id ;
	String subjects[ ];
	String contents[ ];
	String Departments ;
	Int salary ;
	Relation:
	ProctorOf;
	Coursewith ;
	}

Fig 5:Class_Types Representation in ORDB

These three representation models and definitions for the Knowledgebases presented in this paper will serve as the backend for an Expert System. These concepts and KUs defined here are focusing on the Expert Education System, which can use these definitions for enabling the improvement in the teaching-learning processes in any learning system, including individual learning to organizational learning.

5. IMPLEMENTATION & RESULTS

teaching-

The implementation of the "Intelligent Educational System-EduNiversity" is carried out using four knowledge representation models, which were part of our earlier research [13].

- Content Parse Model (CPM)
- Object Structured Model (OSM)
- Concept Structured Model (CSM)
- ConceptFunction Relation Model (CFRM)

To carry out this validation the Structure-Component-Connection (SCC) approach [14]is adopted. Table 2 shows the details and specifications of KB built as a model.

Table 2. Data Collection in SCC approach

Model	Structures built	Components integrated	Connections
CPM	7	22	11
OSM	15	46	24
CSM	6	18	21
CCM	12	29	17
Knowledgebase	2	8	12

The proposed models are codified in Java and C++ language and the backend used to store the instances of these models is MySQL. The number of structures built in Java/C++ platforms, components integrated with the structure, and the connections with the external structures is given in numbers.

The knowledgebase designed is reliable in its existence in a deployable platform. The knowledgebase schema designed consists of class_types, tables, relations, and objects instantiated correspondingto each model are given in Table 3.

Table 3. Experimental Setup for evaluating reliability

Model	Relational Tables	Class Types	Relations	Objects Instantiated (approx.)
CPM (acquisition)	4	28	3	300
OSM (Representation)	8	62	4	650
CSM (Representation)	2	11	16	130
CFRM (Storage)	5	20	4	200
Knowledgebase	12	174	22	1400

5.1 Performance

While the performance of a system covers various factors, the performance transaction throughput, no. of hits to the KB, and no. of requests serviced, all against a time frame of 1 – 180 secs in a given test scenario are opted. Table 4 shows the collated performance test results.

Table 4. Performance Throughput of the knowledgebase

Models	Performance Throughput in (kbps)		
	Max	Min	Average
CPM	78	3	39.57143
OSM	102	5	57.14286
CSM	58	2	27.85714
CFRM	73	3	36.42857
Knowledgebase	33	4	16.28571

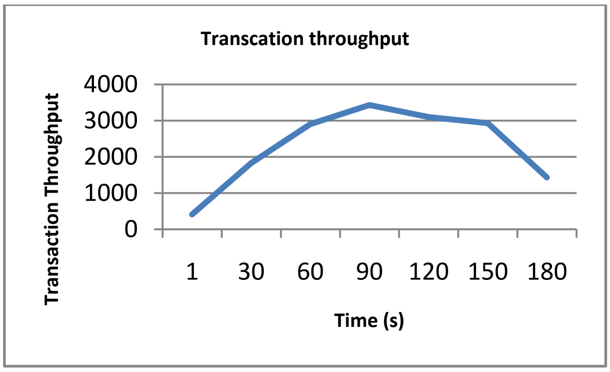

Fig 6: Transaction Throughput

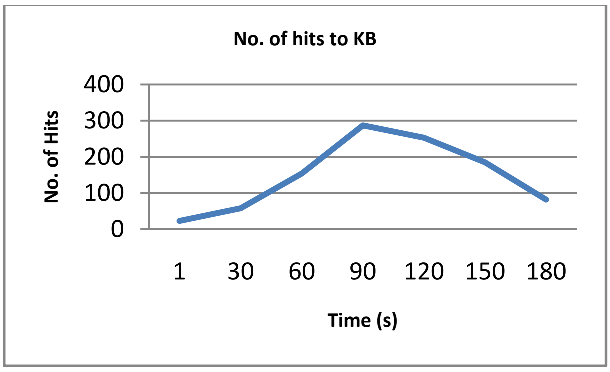

Fig 7: No. of Hits to the KB

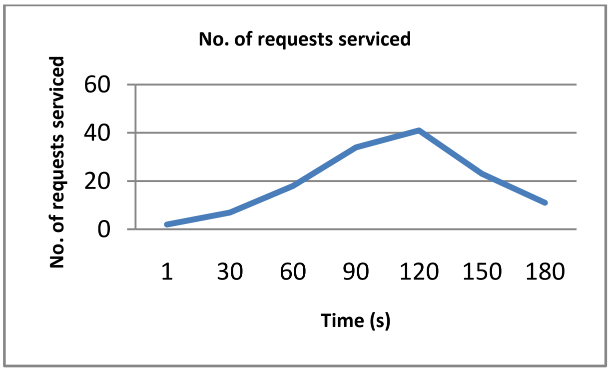

Fig 8: No. of Requests serviced by the KB

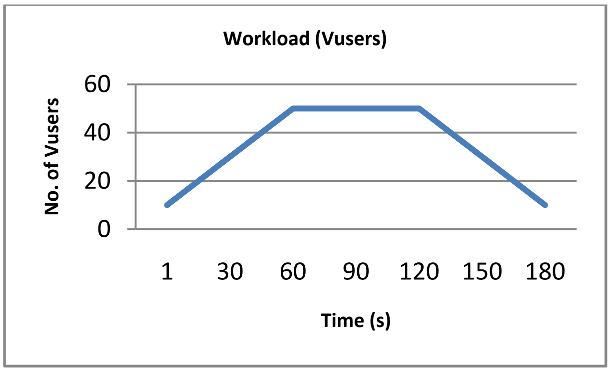

Fig 9: Workload given to the KB

These individual performance test evaluation reports show that the workload, shown in Figure 9, in terms of the virtual users assignment to the KB is steadily ramped up for a period of 60secs and a steady workload is maintained for over another 60mins and then the ramped down over the next 60mins. The corresponding results for the transaction throughput are given in Fig. 6, which shows proportionality in the workload alone, and no bottlenecks are encountered in this controlled environment. The other two parameters of measures are shown in Fig. 7, and Fig. 8, all of them showing an equivalent proportionality to the number of Vusers in the Test scenario over a given period of time.

5.2 Scalability

Scalability, in this context,is defined as the extensibility of the design and definition of the proposed models and the knowledgebase itself. Hencescalability is evaluated in three facets: design-level, functional-level, and storage-level using the models. The results are given in Table 5.

Table 5. Scalability at Design-Level

Models	Concepts		Relations		Objects	
	Prelim	Main	Prelim	Main	Prelim	Main
CPM	14	48	26	82	160	2000
OSM	22	86	47	180	3010	7000
CSM	20	82	70	320	2550	6300
CFRM	500	3200	1320	5400	4000	10200

The scalability was tested at the three layers: Design layer, Functional level, and Storage layers. Throughput was the measurement criteria for testing scalability of the KB design.

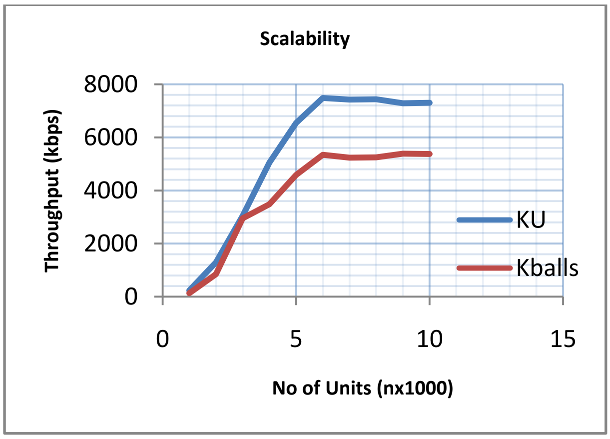

Fig 10: Throughput for KU and KBalls in Part A

In Fig. 10, the number of KUs (knowledge units) and KBalls (knowledge objects) in horizontal axis versus the number of kilo-bytes transacted in a given time in vertical axis generated this graph. The findings are found to be promising during this implementation of the design of KB for EduNiversity.

In most situations, if the scalable (sustained) performance is measured against growing workloads in terms of volume and complexity, the Teradata platform architecture will definitely provide a more robust scale-up and scale-out model for the KB and its dependent applications

6. CONCLUSIONS

This research work focused on designing a knowledgebase for an Expert Education System (EES). The knowledgebase is proposed as a three-layered architecture – design, knowledge, and storage layers. The concepts acquired are identified as clauses, represented in FOL, and are added to the design layer of KB. These concepts are converted to knowledge units (KUs) by adding rules and relations to the clauses, and they are defined in frames and description language, making the KUs dynamic in runtime. Finally, these KUs are codified to a class_type for making the KUs persistent objects in ORDB.

This KB proposed here can be used by an EES for resolving a given pragmatic real-world situation, computing entailments to add new knowledge to the KB. By such a KB, a user of EES will post queries like “I need Java study materials”, and he will be provided with what he requires, along with info like, who is the owner of the materials, and other related materials on Java for learning.

Future works:

1. Propose an integrated framework for KB and EES
2. Works on entailments to find the actual and relevant knowledge units
3. Evaluate the performance of the KB

This research will provide directions for researchers and scholars to carry out the works in the knowledgebase and engineering the knowledge representations.

7. REFERENCES

[1] Philippe Martin, "Knowledge representation in RDF/XML, KIF, Frame-CG and Formalized-English", Distributed System Technology Centre, QLD, Australia, July 15–19, 2002

[2] Hien D. Nguyen ,1,2Nhon V. Do, Et. Al., Some Criteria of the Knowledge Representation Method for an Intelligent Problem Solver in STEM Education, Applied Computational Intelligence and Soft Computing, Vol. 2020, No. 1, pp. 1-14, May 2020

[3] Randall Davis, "What Is a Knowledge Representation?", Association for the Advancement of Artificial Intelligence, Vol. 14, No. 1, 1993.

[4] Lin, J., Zhao, Y., Huang, W. et al. Domain knowledge graph-based research progress of knowledge representation. Neural Comput&Applic 33, 681–690 (2021).

[5] Ronald J. Brachman, Hector J. Levesque, Knowledge Representation and Reasoning, Morgan Kaufmann, 2004

[6] Chein, M., Mugnier, M.-L.,Graph-based Knowledge Representation: Computational Foundations of Conceptual Graphs, Springer, 2009.

[7] Y. Wang and M. Valipour, “Formal ontology generation by deep machine learning,” in Proceedings of 2017 IEEE 16th International Conference on Cognitive Informatics &CognitiveComputing (ICCI∗CC), Oxford, UK, July 2017.

[8] Hermann Helbig: Knowledge Representation and the Semantics of Natural Language, Springer, Berlin, Heidelberg, New York 2006

[9] N. V. Do, H. D. Nguyen, and A. Selamat, “Knowledge-based model of expert systems using rela-model,” International Journal of Software Engineering and Knowledge Engineering, vol. 28, no. 8, pp. 1047–1090, 2018.

[10] John F. Sowa: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole: New York, 2000

[11] Amjad Farooq and Imran Mir, “Design and Development of a University Web Tools”, International journal of multidisciplinary sciences and engineering, vol. 1, no. 2, 2010.

[12] Christy Dharmasingh, Justus S., “A Prediction Model addressing the Changes and Challenges in Adopting SEaaS Model during COVID-19”, International Journal of Scientific Research in Computer Science and Engineering, Vol. 9, Issue 4, pp. 14-20, Aug 2021.

[13] Hepsiba Mabel., Justus S., “Performance Assessment of 3-Layered Knowledgebase and Representation Models: An Empirical Study”, International Journal of Knowledge Engineering and Data Mining, Vol. 3, No. 4, pp. 239-254, Dec 2015

[14] Gartland-Jones. A., and P. Copley, “The Suitability of Genetic Algorithms for Musical Composition”, Contemporary Music Review, Vol. 22, No. 3, pp. 43–55, 2003.

8. AUTHOR PROFILE

Dr. Justus Selwyn received his doctorate degree in computer science specializing in software and knowledge engineering. His research specializations include Object-relational data modeling, knowledge engineering, SEaaS, and artificial intelligence.

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 also has practitioner’s experience while working with Software development companies in India.

He is a member of IEEE, ISTE, and IAENG professional associations. He has served as research & project coordinator for PG studies and has been the chair for the software engineering research group at Engineering Institutes. Presently he is working as a Professor in the department of computer science, John Brown University, Arkansas, USA.